

dossier

COMMODORE



**PROGRAMMI
TRUCCHI
DIDATTICA**

COMMODORE

dossier

Mensile
di giochi e programmi
per COMPUTER
COMMODORE

Direttore Responsabile
Marco Bindi

Direttore Tecnico:
Gianroberto Negri

Edizioni **CO.STA** srl
via Mazzini, 128
56100 PISA

Registrazione al Tribunale
di Pisa n. 2 del 22/2/84
Spediz. in abbon. post.
gruppo III/70

COMMODORE DOSSIER

il mensile per realizzare la tua Biblioteca
di Software in tre sezioni:

- Giochi
- Programmi personali
- Programmi di utilità

Lire 2000
I numeri arretrati, a lire 4000, possono
essere richiesti senza oneri di spedizione
con pagamento anticipato in contanti,
o tramite assegno o vaglia postale
intestato alla Società editrice.

Stampa:
Rotolito Lombarda Spa (MI)

Distribuzione
Messengerie Periodici (MI)

COMMODORE 64: SIMBOLI GRAFICI

SIMBOLO LISTATO

TASTI DA PREMERE

[SPZ]	BARRA SPAZIATRICE
[BNC]	CTRL+2
[GIU']	CRSR(IL PRIMO A SINISTRA)
[RVS ON]	CTRL+9
[HOME]	HOME
[ROSS]	CTRL+3
[DESTR]	CRSR(QUELLO A DESTRA)
[VRD]	CTRL+6
[BLU]	CTRL+7
[ST.]	TASTO CON RAFFIGURATA LA STERLINA ()
[SHFT *]	SHIFT+*
[SHFT +]	SHIFT + IL TASTO +
[CMD -]	COMMODORE + IL TASTO -
[SHFT -]	SHIFT + IL TASTO -
[PI]	IL TASTO PIGRECO
[CMD *]	COMMODORE + *
[F1]	TASTO DI FUNZIONE F1
[F2]	TASTO DI FUNZIONE F2
[F3]	TASTO DI FUNZIONE F3
[F4]	TASTO DI FUNZIONE F4
[F5]	TASTO DI FUNZIONE F5
[F6]	TASTO DI FUNZIONE F6
[F7]	TASTO DI FUNZIONE F7
[F8]	TASTO DI FUNZIONE F8
[NERO]	CTRL + 1
[SU]	SHIFT + CRSR(IL PRIMO A SINISTRA)
[REV OFF]	CTRL + 0
[CLR]	SHIFT + HOME
[INST]	SHIFT + DEL
[PORP]	CTRL + 5
[SIN.]	SHIFT + CRSR(QUELLO A DESTRA)
[GIA]	CTRL + 8
[CYN]	CTRL + 4
[SHFT SPZ]	SHIFT + BARRA SPAZIATRICE
[CMD ST.]	TASTO COMMODORE + TASTO STERLINA ()
[SHFT ST.]	SHIFT + TASTO STERLINA ()
[SHFT @]	SHIFT + @
[DEL]	DEL

NEL NUMERO DI LUGLIO/AGOSTO DI PERSONAL COMPUTER

**Le prove su strada:
Spectravideo 728 e PC Integral**

**Programma/Mese:
Buone vacanze con C64, Apple, PC Ibm, Spectrum e MSX**

**E gli speciali:
TUTTOCOMMODORE - TUTTAPPLE - TUTTOMSX
SINCLAIRDOSSIER - TEXASDOSSIER**

Con l'insero
**STUDIARE
INFORMATICA IN ITALIA**



SPRITE 3

In questo numero, come promesso vedremo come ingrandire le SPRITE e come controllarle quando entrano in collisione.

Iniziamo perciò dall'ingrandimento delle SPRITE e per comprendere come realizzarlo, facciamo capo al programma sottoriportato.

```
1 RESTORE
5 M=832
10 FOR X= 0 TO 62 : READ N : POKE M+X,N : NEXT X
20 POKE 2040,13
30 S=53248
40 POKE 53280,0:POKE53281,0
50 POKE S+21,PEEK(S+1) OR 1
60 POKE S+39,3
70 POKE S+1,30
```

```
-----
80 POKE S+23,1
90 FOR A= 1 TO 255 : POKE S+0,A: NEXT A
95 FOR T=1 TO 200:NEXT T
```

```
-----
100 POKE S+29,1
110 FOR A= 1 TO 255 : POKE S+0,A: NEXT A
115 FOR T=1 TO 200:NEXT T
```

```
-----
120 POKE S+23,0
130 FOR A= 1 TO 255 : POKE S+0,A: NEXT A
135 FOR T=1 TO 200:NEXT T
```

```
-----
140 POKE S+29,0
150 FOR A= 1 TO 255 : POKE S+0,A: NEXT A
155 FOR T=1 TO 200:NEXT T
```

```
-----
160 END
```

```
-----
1000 DATA 0,0,0,0,0,0,0,0,0
1010 DATA 0,0,0,0,0,0,0,0,0
1020 DATA 7,255,0,15,255,128
1030 DATA 15,255,192,31,255,224
1040 DATA 31,255,240,63,255,248
1050 DATA 127,255,255,127,255,0
1060 DATA 0,255,0,0,255,0,0,0
1070 DATA 0,0,0,0,0,0,0,0,0,0,0
```

Tralasciamo le istruzioni che vanno dalla 1 alla 70 e

che si occupano di gestire la creazione e l'inizializzazione della SPRITE, perche' gia' ampiamente spiegate nelle precedenti puntate, e soffermiamoci sulle istruzioni che vanno dalla 80 in poi.

Istruzione 80: in essa viene caricato nel REGISTRO 53271 (poiche' nella variabile S e' contenuto il valore 53248, la somma di S + 23, da' appunto 53271), mediante la parola chiave POKE, il numero 1.

E' questa l'istruzione che fa comprendere al COMMODORE 64 che vogliamo che venga allargata in senso verticale la SPRITE. Il registro 53271 e' generalizzato, non e' cioe' possibile espandere una SPRITE, indipendentemente dalle altre; una volta che il comando e' stato dato, tutte le SPRITE presenti in quel momento sul video ed anche quelle che verranno posizionate successivamente, saranno sempre allargate in senso verticale (asse Y).

Istruzione 90: viene spostata, mediante le parole chiave FOR, TO, POKE e NEXT, la SPRITE numero 0, precedentemente ingrandita in senso verticale, dal punto 1 al punto 255.

Istruzione 95: viene generato un ciclo di ritardo, mediante le parole chiave FOR, TO e NEXT, da 1 a 200.

Istruzione 100: in essa viene caricato nel REGISTRO 53277 (poiche' nella variabile S e' contenuto il valore 53248, la somma di S + 29, da' appunto 53277), mediante la parola chiave POKE, il numero 1.

E' questa l'istruzione che fa comprendere al COMMODORE 64 che vogliamo che venga allargata in senso orizzontale la SPRITE. Anche il registro 53277, come il registro 53271 e' generalizzato, non e' cioe' possibile espandere una SPRITE, indipendentemente dalle altre; una volta che il comando e' stato dato, tutte le SPRITE presenti in quel momento sul video ed anche quelle che verranno posizionate successivamente, saranno sempre allargate in senso orizzontale (asse X).

Istruzione 110: viene spostata, mediante le parole chiave FOR, TO, POKE e NEXT, la SPRITE numero 0, precedentemente ingrandita anche in senso orizzontale, dal punto 1 al punto 255.

Istruzione 115: viene generato un ciclo di ritardo, mediante le parole chiave FOR, TO e NEXT, da 1 a 200.

Istruzione 120: in essa viene caricato nel REGISTRO 53271 (poiche' nella variabile S e' contenuto il valore 53248, la somma di S + 23, da' appunto 53271), mediante la parola chiave POKE, il numero 0.

E' questa l'istruzione che fa comprendere al COMMODORE 64 che vogliamo che venga disattivata in senso verticale la SPRITE. Viene fatta cioe' ritornare alla normale altezza. Una volta che il comando e' stato dato, tutte le SPRITE presenti in quel momento sul video ed anche quelle che verranno posizionate successivamente, ritorneranno alle normali dimensioni verticali (asse Y).

Istruzione 130: viene spostata, mediante le parole chiave FOR, TO, POKE e NEXT, la SPRITE numero 0, precedentemente fatta tornare alle normali dimensioni verticali, dal punto 1 al punto 255.

Istruzione 135: viene generato un ciclo di ritardo, mediante le parole chiave FOR, TO e NEXT, da 1 a 200.

Istruzione 140: in essa viene caricato nel REGISTRO 53277 (poiche' nella variabile S e' contenuto il valore 53248, la somma di S + 29, da' appunto 53277), mediante la parola chiave POKE, il numero 0.

E' questa l'istruzione che fa comprendere al COMMODORE 64 che vogliamo che venga disattivata in senso orizzontale la SPRITE. Viene fatta cioe' ritornare alla normale lunghezza una volta che il comando e' stato dato; tutte le SPRITE presenti in quel momento sul video ed anche quelle che verranno posizionate successivamente, ritorneranno alle normali dimensioni orizzontali (asse X).

Istruzione 150: viene spostata, mediante le parole chiave FOR, TO, POKE e NEXT, la SPRITE numero 0, precedentemente fatta tornare alle normali dimensioni orizzontali, dal punto 1 al punto 255.

Istruzione 155: viene generato un ciclo di ritardo, mediante le parole chiave FOR, TO e NEXT, da 1 a 200.

Istruzione 160: viene fermata l'esecuzione del programma.

Ricapitolando il tutto: le istruzioni che vanno dalla 80 alla 95 allargano la SPRITE in senso VERTICALE e la fanno muovere, orizzontalmente sul video, dalla posizione 1 alla posizione 255.

Le istruzioni che vanno dalla 100 alla 115 allargano la SPRITE in senso ORIZZONTALE e la fanno muovere, orizzontalmente sul video, dalla posizione 1 alla posizione 255.

Le istruzioni che vanno dalla 120 alla 135 disattivano la SPRITE in senso ORIZZONTALE e la fanno muovere, orizzontalmente sul video, dalla posizione 1 alla posizione 255.

Le istruzioni che vanno dalla 140 alla 145 disattivano la SPRITE in senso VERTICALE e la fanno muovere, orizzontalmente sul video, dalla posizione 1 alla posizione 255.

Semplice non trovate?... Vediamo ora di analizzare come gestire le COLLISIONI che si possono verificare tra SPRITE e SPRITE e tra le SPRITE ed i CARATTERI STANDARD.

Come nel caso prima analizzato, anche per comprendere come gestire le COLLISIONI, ricorriamo ad un programma, ed andiamo ad analizzare le sue istruzioni e parole

chiave.

2 RESTORE

8 M=832

15 FOR X= 0 TO 62 : READ N : POKE M+X,N : NEXT X

25 POKE 2040,13

33 S=53248

45 POKE 53280,0:POKE53281,0

50 POKE S+21,PEEK(S+1) OR 1

60 POKE S+39,3

70 POKE S+1,50

80 FOR A= 1 TO 155 : POKE S+0,A: NEXT A

90 POKE S+21,PEEK(S+1) OR 16

92 POKE S+43,2

93 POKE S+8,155

94 FOR A= 250 TO 50 STEP -1: POKE S+9,A: NEXT A

100 IF PEEK(S+30)<>0 THEN POKE S+21,PEEK(S+21) AND 239:
POKE S+21,PEEK(S+21) AND 254

110 DF=PEEK(S+30)

120 END

2000 DATA 0,0,0,0,0,0,0,0,0,0

2010 DATA 0,0,0,0,0,0,0,0,0,0

2020 DATA 7,255,0,15,255,128

2030 DATA 15,255,192,31,255,224

2040 DATA 31,255,240,63,255,248

2050 DATA 127,255,255,127,255,0

2060 DATA 0,255,0,0,255,0,0,0,0

2070 DATA 0,0,0,0,0,0,0,0,0,0,0

Vediamo di riassumere brevemente le istruzioni di supporto, soffermandoci solo su quelle direttamente interessate alla gestione delle collisioni.

Istruzione 2: mediante la parola chiave RESTORE, viene ripristinata la lettura dei valori riportati nelle istruzioni che vanno dalla 2000 in poi.

Istruzione 8: viene caricato nella variabile M il punto d'inizio delle memoria, nella quale caricare i valori riportati nelle istruzioni che vanno dalla 2000 in poi.

Istruzione 15: vengono letti i valori dalle istruzioni comprese tra la 2000 e la 2070, caricandoli in memoria, dalla locazione 832 in poi.

Istruzione 25: viene comunicato al COMMODORE 64 il punto iniziale al quale si trovano i valori da utilizzare per creare le SPRITE utilizzate nel programma.

Istruzione 33: viene caricato nella variabile S, il REGISTRO di partenza adibito alla gestione delle SPRITE.

Istruzione 45: vengono settati i colori del fondo e del bordo.

Istruzione 50: in essa viene attivata la SPRITE numero 0, mediante le parole chiave POKE e PEEK.

Istruzione 60: viene assegnato il colore numero 3 alla SPRITE numero 0, corrispondente al CYAN.

Istruzione 70: viene posizionata al punto 50 dell'asse Y (VERTICALE) la SPRITE numero 0.

Istruzione 80: viene spostata, mediante le parole chiave FOR, TO, POKE e NEXT, la SPRITE numero 0, dal punto 1 al punto 155.

Istruzione 90: in essa viene attivata la SPRITE numero 4, mediante le parole chiave POKE e PEEK.

Istruzione 92: viene assegnato il colore numero 2 alla SPRITE numero 0, corrispondente al ROSSO.

Istruzione 93: viene posizionata al punto 155 dell'asse X (VERTICALE) la SPRITE numero 4.

Istruzione 94: viene spostata, mediante le parole chiave FOR, TO, POKE, NEXT e STEP, la SPRITE numero 4, dal punto 250 al punto 50.

Istruzione 100: mediante le parole chiave IF, PEEK e THEN viene controllato se nella LOCAZIONE di memoria S+30, (corrispondente a: 53278 poiche' nella variabile S e' contenuto il valore 53248, la somma di S + 30, da' appunto 53278) vi sia contenuto un valore numerico diverso da 0.

In caso affermativo mediante le parole chiave POKE, PEEK e AND vengono disattivate le SPRITE 0 e 4.

La locazione di memoria 53278 e' quella adibita al rilevamento delle collisioni tra SPRITE e quando contiene un valore numerico maggiore di zero, significa che due o piu' SPRITE sono entrate in collisione.

Essa presenta anche una caratteristica da tenere in dovuto conto: ogni volta che viene effettuata una lettura mediante la parola chiave PEEK, il suo contenuto viene automaticamente azzerato.

Nel caso di collisioni tra due SPRITE, come nel programma sopra riportato, il fatto non costituisce grossi problemi; nel caso che pero' siano avvenute collisione tra piu' di due SPRITE occorre salvare il contenuto della locazione di memoria in una variabile, per poterlo poi riutilizzare e determinare

quale altre sprite sono entrate in collisione.

Istruzione 110: essa e' stata inserita in qualita' di trucco da utilizzare nella realizzazione d'animazioni e/o giochi ed ha questo preciso scopo: azzerare realmente il registro 53278. In che senso realmente?... Nel senso che (anche se in questo caso, non viene utilizzato) durante lo svolgimento del gioco, se dopo che abbiamo testato la presenza o meno di collisioni, non spostiamo le SPRITE, nel registro sara' contenuto un valore maggiore di zero, segno che le SPRITE sono sempre in collisione. Quindi il trucco consiste nel: rilevare una prima volta se si sono verificate delle collisioni, in caso affermativo, gestire i vari effetti contemplati nel gioco, spostare le SPRITE ed azzerare, mediante la parola chiave PEEK, una seconda volta il registro che nel frattempo avrebbe rilevato un'altra collisione data dal fatto che dopo il primo rilevamento, prima che sia terminata la parte di programma adibita alla collisione, le SPRITE non erano ancora state spostate. Per concludere, questo piccolo artificio permette d'eliminare le false rilevazioni di collisione.

Istruzione 120: mediante la parola chiave END viene fatta fermare l'esecuzione del programma.

Questo per quanto riguarda le collisioni avvenute tra SPRITE e SPRITE, nel caso invece di collisioni tra SPRITE e CARATTERI STANDARD, il controllo e le problematiche, sono le medesime, solo che il registro interessato e' il 53279. Detto questo e' giunto il momento di salutarci sino al prossimo numero della rivista, nella quale analizzeremo finalmente un completo programma di un gioco, in cui utilizzeremo tutte le tecniche sino a questo punto analizzate piu' diversi trucchi atti a rendere il gioco piu' semplice a gestirsi; a presto.



INPUT

Terza puntata di questa interessante analisi del BASIC e delle sue PAROLE CHIAVE. In questo numero

vederemo d'utilizzare le VARIABILI, delle quali, abbiamo parlato nella precedente puntata ricordate?

L'utilizzo piu' CLASSICO che si puo' fare delle VARIABILI e' unitamente alla PAROLA CHIAVE INPUT.

Sempre nella puntata precedente avevamo detto che il linguaggio BASIC era nato nei paesi anglosassoni e

tutti noi sappiamo che la lingua ufficiale di tali paesi e' l'INGLESE. Quindi le parole chiave del BASIC sono sempre espresse appunto in INGLESE, per cui la parola chiave INPUT se viene tradotta in italiano corrente significa: ENTRA. A che cosa serve?... Vediamo di fare un esempio atto a chiarificare il tutto, per cui ipotizziamo di voler immettere o comunicare dei DATI al nostro fido computer. Come poterlo fare?... Come poter far comprendere ad esso di fermare cio' che in quel momento sta facendo, in

maniera da potergli battere dei dati da tastiera?... Se volessimo comunicare delle informazioni ad un nostro caro amico, basterebbe dirgli: "ASPETTA UN ATTIMO TI DEVO COMUNICARE UNA COSA!..." ed il nostro amico a questo punto rimarrebbe in attesa di quanto stiamo per comunicargli. Bene quello che abbiamo appena fatto, anche se apparentemente non sembra, e' di dare un ORDINE di fermarsi e d'ascoltare al nostro amico. Quindi per far fermare il computer, in attesa che gli comunichiamo o trasmettiamo i nostri dati o le nostre informazioni dovremo utilizzare un COMANDO!... Come gia' detto nelle precedenti puntate un programma e' composto da un certo numero d'ISTRUZIONI, ed ogni ISTRUZIONE e' composta da un certo numero di COMANDI. Ed i comandi altro non sono che le PAROLE CHIAVE. Quale parola chiave fa quindi al nostro caso per far fermare il computer, in attesa che gli vengano comunicati dei dati?... La parola chiave INPUT che come gia' detto significa ENTRA. ENTRA che cosa?... Dei dati, delle informazioni, dei numeri, ecc..

Ovviamente una volta immessi o comunicati, i dati devono essere conservati da qualche parte, per poterli utilizzare in un tempo successivo, perche' in caso contrario andrebbero persi. Ecco quindi che vengono in nostro aiuto le VARIABILI. E' in esse che vengono salvati e conservati i dati. Praticamente come delle matite (i dati) che vengono conservate in dei contenitori (le variabili), pronte ad essere riutilizzate, quando se ne presenta la necessita'. Per poterla correttamente utilizzare, la parola chiave INPUT deve essere cosi' disposta entro l'istruzione:

10 INPUT A

E il numero 10, posto prima della parola chiave INPUT, che significato ha e a che cosa serve?... Vi chiederete... Per poterlo spiegare facciamo un attimo un salto in una cucina, ed immaginiamo d'essere dei famosi cuochi, impegnati nella realizzazione di una ricetta, segretamente passataci da un nostro collega. La ricetta e' composta da una serie d'istruzioni poste in sequenza, cioe' una di seguito all'altra. Affinche' la pietanza (il piatto) riesca e' necessario seguire quanto in essa e' descritto e soprattutto di rispettarne la corretta sequenza. Se volessimo eseguire: prima quando detto in fondo alla ricetta e poi quando detto all'inizio, tutto verrebbe fuori, fuorché il piatto descritto dalla ricetta. Per questo motivo, molte volte, le istruzioni di cui e' composta la ricetta sono numerate, affinche' non si sbagli la sequenza con cui esse vanno eseguite. Quindi, esaminandola, troveremo prima l'istruzione 1, poi la 2, poi la 3 e cosi' via, sino alla fine della ricetta.

Il numero posto prima della parola chiave INPUT serve al medesimo scopo: far comprendere al computer l'esatta sequenza con cui vogliamo vengano

eseguiti i comandi che gli abbiamo impartito. Semplice!... Non trovate?...

La lettera A, posta dopo d'essa e', come avrete gia' capito, una VARIABILE e per essere piu' precisi una VARIABILE di tipo REALE. I dati immessi mediante la parola chiave INPUT, verranno conservati in essa, pronti ad essere utilizzati quando lo si desidera. Ovviamente essendo nell'esempio riportato, presente una sola VARIABILE, potra' essere immesso un solo dato numerico.

Nel caso si desiderasse immettere piu' di un dato, ad esempio tre dati, l'istruzione sopra riportata dovra' essere cosi' modificata:

10 INPUT A,B,C

Questo per quanto riguarda dati di tipo numerico; nel caso invece di dati di tipo alfanumerico al posto delle VARIABILI di tipo numerico REALE, dovranno essere utilizzate delle VARIABILI di tipo ALFANUMERICO ed in tal caso l'istruzione diverrebbe:

10 INPUT A\$,B\$,C\$

Rimane ovvio che e' possibile abbinare nella medesima istruzione, dopo la parola chiave INPUT, diversi tipi di variabili, come nell'istruzione di seguito riportata:

10 INPUT A\$,A

Vediamo di commentarla dettagliatamente:

La parola chiave INPUT fa fermare l'esecuzione del programma e fa attendere il nostro fido computer sino al momento che vengono immessi dei dati ed in questo caso potrebbero ad esempio essere il nostro indirizzo ed il nostro numero telefonico.

I due dati dovranno essere separati da una virgola come nell'esempio riportato qui sotto:

via Tosi 22,340899

Per far comprendere che abbiamo finito dovra' essere premuto il tasto RETURN, quale conferma dell'avvenuta immissione. A questo punto VIA TOSI 22 corrispondente al nostro indirizzo sara' caricato nella variabile di tipo alfanumerico A\$ e 340899, corrispondente al nostro numero di telefono, sara' caricato nella variabile di tipo numerico reale A. Questi due dati saranno in tal modo a nostra disposizione, sino al momento che desidereremo utilizzarli.

Il tasto RETURN e' un tasto estremamente importante, poiche' soltanto mediante esso potremo comunicare al computer che abbiamo terminato quanto in quel momento stavamo facendo.

Questo vale sia nel caso di caricamento d'istruzioni, sia nel caso d'immissione di dati.

Prima di concludere, un'altra importante precisazione: Una volta che l'istruzione è stata immessa ed è stato battuto il tasto RETURN, quale conferma, dovrà essere battuta la parola chiave RUN e dovrà essere premuto il tasto RETURN. Solo in questo modo potrà essere attivato il programma sino a quel momento immesso. Su di essa è doveroso precisare una sua importante caratteristica: l'azzeramento di tutte le variabili contenute nel programma. Quindi ogni volta che l'istruzione 10 del nostro esempio, sarà attivata ed il computer inizierà ad eseguirla, quanto in quel momento era contenuto nelle variabili A\$ ed A, verrà irrimediabilmente perduto. Quindi il nostro indirizzo ed il nostro numero di telefono ogni volta

che verrà dato il RUN, andranno persi. Questa caratteristica è di estrema importanza e, dovrà essere sempre tenuta presente se non si vuole perdere dati di primaria importanza, nel caso che il programma si fermi a causa di qualche errore!!... In una delle prossime puntate vedremo come aggirare questo problema e riuscire lo stesso ad attivare il programma, senza per questo perdere i dati contenuti, entro le sue variabili. Per i più impazienti diremo che lo si potrà fare, ricorrendo ad una parola chiave del BASIC che... Prendete i prossimi numeri e lo scoprirete... Per questa volta Vi salutiamo, e a rileggerci sul prossimo numero.



TRAP

Nei precedenti numeri abbiamo visto l'utilizzo della parola chiave USING, unitamente alla parola chiave PRINT. Essa trova ampio utilizzo nelle applicazioni di tipo GESTIONALE per formattare i dati da visualizzare e/o stampare. Un'altra parola chiave di ampio utilizzo nelle applicazioni a carattere GESTIONALE o professionale è la parola chiave TRAP. Prima però d'addentrarci nella sua dettagliata analisi, vediamo di comprendere il perché sia stata inclusa nel BASIC 3.5 del COMMODORE 16. Per far ciò, come ormai nostra consuetudine, facciamo riferimento, al programma sottoriportato ed analizziamo il funzionamento istruzione per istruzione:

```
10 INPUT "IMMETTERE PRIMO VALORE ";A
20 INPUT "IMMETTERE SECONDO VALORE ";B
30 PRINT "ECCO IL RISULTATO: ";A/B
```

L'istruzione 10, mediante la parola chiave INPUT stampa a video il messaggio "IMMETTERE PRIMO VALORE ", fa fermare l'esecuzione del programma in attesa che venga immesso un dato per caricarlo nella variabile di tipo numerico reale A.

L'istruzione 20, mediante la parola chiave INPUT, stampa a video il messaggio "IMMETTERE SECONDO VALORE ", fa fermare l'esecuzione del programma, in attesa che venga immesso un dato per caricarlo nella variabile di tipo numerico reale B.

L'istruzione 30, mediante la parola chiave PRINT,

stampa a video il messaggio "ECCO IL RISULTATO: ", effettua la divisione del numero contenuto nella variabile di tipo numerico reale A per la variabile di tipo numerico reale B, stampandolo a video.

Se nella variabile di tipo numerico reale B, si trova un numero maggiore di zero, non vi sono problemi di sorta ed il risultato della divisione viene stampato a video mediante la parola chiave PRINT dell'istruzione 30.

Nel caso però che in essa venga immesso il numero zero, le cose si complicano essendo una divisione di un numero per zero non consentita!!... Il risultato sarebbe la visualizzazione del messaggio di sistema:

?DIVISION BY ZERO ERROR

Il programma si interromperebbe e per riutilizzarlo, bisognerà riattivarlo, mediante la parola chiave RUN, con la conseguente perdita di quanto contenuto nelle variabili del programma.

Se vi fosse un metodo per intercettare GLI ERRORI, senza far fermare l'esecuzione del programma, con la conseguente perdita dei dati, il tutto sarebbe molto più semplice ed efficace!!...

Ecco intervenire in nostro aiuto la parola chiave TRAP, progettata proprio per permetterci di gestire gli errori senza per questo far fermare l'esecuzione del programma.

Vediamo d'aggiungere l'istruzioni necessarie, al programma visto prima, in modo da gestire l'errore che si manifesta se cerchiamo di dividere un numero per zero.

La nuova versione del programma e' questa:

```
5 PRINT CHR$(147);
6 TRAP 1000
8 OR=0
10 INPUT "IMMETTERE PRIMO VALORE ";A
20 INPUT "IMMETTERE SECONDO VALORE ";B
30 PRINT "ECCO IL RISULTATO: ";A/B
45 IF OR=1 THEN GOTO 8
50 END
1000 PRINT CHR$(147);:PRINT" DIVISIONE PER ZERO NON VALIDA"
1005 OR=1
1010 RESUME NEXT
```

Istruzione 5: mediante la parola chiave PRINT ed il carattere corrispondente al codice 147, viene cancellato quanto in quel momento e' presente sul video.

Istruzione 6: mediante la parola chiave TRAP, nel caso si verificasse un errore, esso viene intercettato e viene effettuato un salto all'istruzione il cui numero e' riportato dopo la parola chiave, ed in questo caso all'istruzione 1000.

Istruzione 8: viene caricato nella variabile di tipo numerico reale OR il valore numerico 0.

Istruzione 10, mediante la parola chiave INPUT stampa a video il messaggio "IMMETTERE PRIMO VALORE ", fa fermare l'esecuzione del programma, in attesa che venga immesso un dato per caricarlo nella variabile di tipo numerico reale A ed in questo caso ipotizziamo che sia il valore numerico 15.

Istruzione 20, mediante la parola chiave INPUT stampa a video il messaggio "IMMETTERE SECONDO VALORE ", fa fermare l'esecuzione del programma, in attesa che venga immesso un dato per caricarlo nella variabile di tipo numerico reale B ed in questo caso ipotizziamo che sia il valore numerico zero.

Istruzione 30: mediante la parola chiave PRINT, stampa a video il messaggio "ECCO IL RISULTATO: ", effettua la divisione del numero contenuto nella variabile di tipo numerico reale A per la variabile di tipo numerico reale B, stampandolo a video.

Il tentativo da parte dell'istruzione 30 di dividere un numero per zero, genera un errore!... Ma a causa della presenza della parola chiave TRAP all'istruzione 6, anziche' visualizzare il messaggio d'errore ed interrompere l'esecuzione del programma, passa il controllo all'istruzione il cui numero e' posto dopo la parola chiave TRAP ed in questo caso all'istruzione 1000.

Istruzione 45: viene controllato, mediante le parole chiave IF e THEN, se nella variabile di tipo NUMERICO REALE OR, vi sia contenuto il valore numerico 1, ed in caso affermativo, mediante la parola chiave GOTO, viene effettuato un salto all'istruzione 8, dove viene azzerato il contenuto della variabile di tipo NUMERICO

RALE OR e viene riproposto d'immettere due numeri di cui effettuare la divisione.

Istruzione 50: mediante la parola chiave END, viene interrotta l'esecuzione del programma.

Istruzione 1000: e' questa la prima istruzione che si occupa di gestire gli errori che si verificano all'interno del programma, e alla quale si perviene mediante la parola chiave TRAP, posta nell'istruzione 6; in essa mediante la parola chiave PRINT ed il carattere di controllo, il cui codice e' 147, viene cancellato quanto in quel momento e' presente sul video. La successiva parola chiave PRINT si occupa, di stampare a video il messaggio: " DIVISIONE PER ZERO NON VALIDA".

Istruzione 1005: viene caricato il valore numerico 1 nella variabile di tipo numerico OR.

Istruzione 1010: mediante le parole chiave RESUME e NEXT, viene fatta continuare l'esecuzione del programma, dall'istruzione successiva a quella in cui si e' verificato l'errore, ed in questo caso l'istruzione 45.

Vediamo ora di riassumere il tutto per quanti nutrissero ancora dei dubbi sul corretto utilizzo della parola chiave TRAP, soffermandoci in particolare sulle istruzioni 8, 45 e 1005.

L'istruzione 5 serve per cancellare quanto in quel momento e' presente sul video.

L'istruzione 6 e' quella in cui e' presente la parola chiave TRAP e l'istruzione alla quale saltare, nel caso si verificasse un errore. Questa istruzione e' di estrema importanza che venga posta all'inizio del programma, perche' in caso contrario, il verificarsi di un errore, non potrebbe essere controllato da programma. Il numero d'istruzione 1000, posto dopo la parola chiave TRAP, e' bene corrisponda alle ultime istruzioni del programma, questo per poter individuare con un solo colpo d'occhio, quanti e quali tipi d'errori sono da essa controllati e poter quindi intervenire con aggiunte e/o modifiche.

L'istruzione 8, in cui viene azzerata la variabile di tipo numerico reale OR, serve per ripristinare la condizione di non errore e quindi di evitare che il controllo effettuato all'istruzione 45, faccia entrare il programma in un ciclo infinito, da cui l'unica possibilita' d'uscita e' rappresentata dal comando STOP, dato premendo il tasto apposito, presente sulla parte sinistra della tastiera del COMMODORE16. Tale ciclo, viene attivato mediante l'istruzione 1005, in cui viene caricato il valore numerico 1 nella variabile di tipo numerico reale OR, al verificarsi di un errore, essendo tale istruzione compresa tra quelle adibite alla gestione degli errori.

Le istruzioni che vanno dalla 10 alla 30, si occupano di richiedere l'immissione dei due numeri da dividere e della stampa degli stessi.

L'istruzione 45, come gia' precisato serve per far entrare il programma in un ciclo che, verra'

disattivato in caso di dati correttamente immessi ed in questo caso: valori numerici maggiori di zero.

L'istruzione 50, serve per interrompere il programma e quindi per evitare che vengano eseguite l'istruzioni che vanno dalla 1000 alla 1010 e che si occupano di gestire gli errori.

Le istruzioni 1000-1010, come già detto si occupano di gestire l'errore che si verifica in caso vengano immessi dei dati uguali a zero.

Nel programma riportato è stato controllato un SOLO tipo di errore, quello cioè che si verifica quando si cerca di dividere un numero per zero; ovviamente i possibili errori che si possono verificare in un programma sono molteplici, e quindi per avere una routine veramente valida è necessario controllarli tutti, o quanto meno la maggior parte.

Ecco quindi come appare il programma, sino ad ora analizzato, dopo le aggiunte atte a controllare la maggior parte degli errori che si possono verificare durante l'esecuzione di un programma.

```
5 PRINT CHR$(147);
6 TRAP 1000
8 OR=0
10 INPUT "IMMETTERE PRIMO VALORE ";A
20 INPUT "IMMETTERE SECONDO VALORE ";B
30 PRINT "ECCO IL RISULTATO: ";A/B
45 IF OR=1 THEN GOTO 8
50 END
1000 PRINT CHR$(147);
1001 OR = 1
1004 IF ER=4 THEN PRINT "FILE NON TROVATO"
1005 IF ER=5 THEN PRINT "DISPOSITIVO NON PRESENTE"
1014 IF ER=14 THEN PRINT "QUANTITA NON VALIDA"
1015 IF ER=15 THEN PRINT "ERRORE DI OVERFLOW"
1020 IF ER=20 THEN PRINT "DIVISIONE PER ZERO NON VAL
IDA"
1023 IF ER=23 THEN PRINT "TROPPI CARATTERI (MAX.255)"
1036 IF ER=36 THEN PRINT "DISCHETTO NON VALIDO"
1050 RESUME NEXT
```

Vediamo di analizzare le modifiche e le aggiunte introdotte:

Istruzione 1001: ha sostituito l'istruzione 1005 e come in essa, viene caricato il valore numerico 1 nella variabile di tipo numerico reale OR.

Istruzione 1004: in essa viene controllato mediante le parole chiave IF e THEN se il valore contenuto nella variabile di tipo numerico reale ER sia 4.

La variabile ER è una VARIABILE DI SISTEMA, cioè una variabile utilizzata dal SISTEMA OPERATIVO per caricarvi dei valori, ed in questo caso per caricarvi il numero dell'errore che si è verificato. Essendo il valore 4, corrispondente all'errore che si verifica quando viene cercato di richiamare un FILE inesistente dal dischetto, il messaggio stampato a video, mediante la parola chiave PRINT, posta dopo la parola chiave

THEN, sempre nell'istruzione 1004 è: "FILE NON TROVATO".

Istruzione 1005: in essa viene controllato mediante le parole chiave IF e THEN se il valore contenuto nella variabile di tipo numerico reale ER sia 5.

Essendo, anche in questa istruzione, la variabile ER una VARIABILE DI SISTEMA utilizzata dal SISTEMA OPERATIVO per caricarvi il numero dell'errore che si è verificato, ed essendo 5 il valore in essa contenuto, viene stampato a video, mediante la parola chiave PRINT, posta dopo la parola chiave THEN, il messaggio: "DISPOSITIVO NON TROVATO", corrispondente al numero dell'errore verificatosi.

Istruzione 1014: in essa viene controllato mediante le parole chiave IF e THEN se il valore contenuto nella variabile di tipo numerico reale ER sia 14.

Essendo anche in questa istruzione, come nella 1005, la variabile ER una VARIABILE DI SISTEMA utilizzata dal SISTEMA OPERATIVO per caricarvi il numero dell'errore che si è verificato, ed essendo 14 il valore in essa contenuto, viene stampato a video, mediante la parola chiave PRINT, posta dopo la parola chiave THEN, il messaggio: "QUANTITA' NON VALIDA", corrispondente al numero dell'errore verificatosi.

Istruzione 1015: in essa viene controllato mediante le parole chiave IF e THEN se il valore contenuto nella variabile di tipo numerico reale ER, utilizzata dal SISTEMA OPERATIVO quale variabile d'errore, sia 15.

Nel caso la condizione si verificasse, viene stampato a video, mediante la parola chiave PRINT, posta dopo la parola chiave THEN, il messaggio: "ERRORE DI OVERFLOW", corrispondente al numero dell'errore verificatosi.

Istruzione 1020: in essa viene controllato mediante le parole chiave IF e THEN se il valore contenuto nella variabile di tipo numerico reale ER, utilizzata dal SISTEMA OPERATIVO quale variabile d'errore, sia 20.

Nel caso la condizione si verificasse, viene stampato a video, mediante la parola chiave PRINT, posta dopo la parola chiave THEN, il messaggio: "DIVISIONE PER ZERO NON VALIDA", corrispondente al numero dell'errore verificatosi.

Istruzione 1023: in essa viene controllato mediante le parole chiave IF e THEN se il valore contenuto nella variabile di tipo numerico reale ER, utilizzata dal SISTEMA OPERATIVO quale variabile d'errore, sia 23.

Nel caso la condizione si verificasse, viene stampato a video, mediante la parola chiave PRINT, posta dopo la parola chiave THEN, il messaggio: "TROPPI CARATTERI (MAX.255)", corrispondente al numero dell'errore verificatosi.

Istruzione 1036: in essa viene controllato mediante le parole chiave IF e THEN se il valore contenuto nella variabile di tipo numerico reale ER, utilizzata dal SISTEMA OPERATIVO quale variabile d'errore, sia 36.

Nel caso la condizione si verificasse, viene stampato a video, mediante la parola chiave PRINT, posta dopo la parola chiave THEN, il messaggio: "DISCHETTO NON

VALIDO", corrispondente al numero dell'errore verificatosi.

Istruzione 1050: mediante le parole chiave RESUME e NEXT viene fatta continuare l'esecuzione del programma, dalla l'istruzione successiva a quella in cui si e' verificato l'errore, ed in questo caso l'istruzione 45.

Ovviamente l'istruzione a cui tornare varia, a secondo del punto dove l'errore viene a verificarsi.

Nei vari esempi riportati, l'istruzione di ritorno e' sempre stata l'istruzione 45, a causa del fatto che in essi il vero e proprio programma e' quello che si occupa di far fermare l'esecuzione del programma, far immettere dei dati, due per l'esattezza, ed effettuare una divisione con gli stessi e di cui l'istruzione 45

ne fa parte integrante.

Il fatto che l'istruzione al quale viene effettuato il salto di ritorno dalla routine che si occupa di gestire gli errori e' la seguente a quella in cui si e' verificato l'errore, va tenuto in debita considerazione.

In caso contrario si corre il rischio di far eseguire delle istruzioni che fanno cose che in realta' non earano state ben preventivate.

La potenza e la versatilita' offerta dalla parola chiave TRAP, unitamente alle parole chiave RESUME e NEXT, permette di realizzare programmi dotati di potenti controlli degli errori, e specialmente programmi che non si fermano all'improvviso, con la conseguente perdita dei dati in quel momento in memoria, a causa di un dato immesso in maniera errata.



#A22 CONTROLLO TASTI

Il contenuto della locazione di memoria 197, determina in ogni momento quale tasto sia stato premuto e puo' essere di aiuto nei Vostri programmi.

Quando nessun tasto e' premuto, il valore numerico in essa contenuto e' 64. L'utilizzo nell'ambito del programma, viene fatto mediante la parola chiave PEEK, nella seguente maniera:

```
10 IF PEEK(197)=64 THEN GOTO 10
20 IF PEEK(197)=(TASTO DA VOI PRESCELTO) THEN...
```

I valori numerici in essa contenuti, variano tra il COMMODORE64 ed il VIC20. La tabella sotto riportata, riporta i valori corrispondenti ai vari tasti, per entrambi i computer.

TASTO	C64	VIC	TASTO	C64	VIC
			:		
A	10	17	: N	39	28
B	28	35	: O	38	52
C	20	34	: P	41	13
D	18	18	: Q	62	48
E	14	49	: R	17	10
F	21	42	: S	13	41
G	26	19	: T	22	50
H	29	43	: U	30	51
I	33	12	: V	31	27

J	34	20	: W	9	9
K	37	44	: X	23	26
L	42	21	: Y	25	11
M	36	36	: Z	12	33
1	56	0	: FRECC.SIN.	57	8
2	59	56	: +	40	5
3	8	1	: -	43	61
4	11	57	: STERLINA	48	6
5	16	2	: @	46	53
6	19	58	: *	49	14
7	24	3	: FRECC.SU	54	54
8	27	59	: :	45	45
9	32	4	: ;	50	22
0	35	60	: =	53	46
,	47	29	: .	44	37
/	55	30	: F1	4	39
SPAZIO	60	32	: F3	5	47
RETURN	1	15	: F5	6	55
STOP	63	24	: F7	3	63
HOME	51	62	: DELETE	0	7
CURS.GIU'	7	31	: CURS.DESTR.	2	23

#A23 DISABILITAZIONE TASTIERA

Vi sono alcune applicazioni in cui si rende necessaria la disabilitazione della tastiera del COMMODORE 64 e del VIC 20. Questo per impedire che vengano premuti tasti che potrebbero pregiudicare il corretto

funzionamento del programma attivo in quel momento.
Per disabilitarla su entrambi occorrerà dare il seguente comando diretto:

POKE 649,0

Per riabilitarla occorrerà invece dare il seguente comando diretto:

POKE 649,10

*A24 DISABILITAZIONE TASTI PARTICOLARI

E' comodo poter disabilitare alcuni tasti come ad esempio: lo STOP, il RUN/RESTORE o la funzione LIST. Questo si rivela molto utile anche quale protezione dei listati da occhi indiscreti o per evitare l'interruzione del programma con la conseguente perdita dei dati contenuti entro le variabili allocate dal programma.

Ecco i comandi da dare in modo diretto o da programma per disabilitare i tasti:

VIC 20:

POKE 808,114 serve a disabilitare il tasto STOP.
La pressione combinata dei tasti RUN/RESTORE lo riabilita.

POKE 808,100 disabilita: STOP, STOP/RESTORE e LIST.
oppure
POKE 808,127

POKE 808,112 ripristina i tasti: STOP, STOP/RESTORE e LIST, disabilitati dalla precedente POKE.

POKE 813,73:POKE 819,245 disabilita il comando SAVE.
La pressione combinata dei tasti STOP/RESTORE lo riabilita.

POKE 775,0 disabilita il LIST

POKE 775,199 riabilita il LIST

COMMODORE 64:

POKE 808,239 serve a disabilitare il tasto STOP.
La pressione combinata dei tasti RUN/RESTORE lo riabilita.

POKE 808,225 disabilita: STOP, STOP/RESTORE e LIST.

POKE 808,237 ripristina i tasti: STOP, STOP/RESTORE e LIST, disabilitati dalla precedente POKE.

POKE 775,200 disabilita il LIST

POKE 775,167 riabilita il LIST

*A25 TASTI DI CONTROLLO

Forse non tutti sanno del possibile utilizzo del tasto CTRL unitamente ad un lettera, oltre al noto utilizzo, insieme ai tasti numerici.

Il suo utilizzo può essere fatto solo entro le virgolette, poste dopo la parola chiave PRINT.

Ecco alcuni utili esempi:

CTRL insieme al tasto H: disabilita i tasti COMMODORE+SHIFT

CTRL insieme al tasto I: riabilita i tasti COMMODORE+SHIFT

CTRL insieme al tasto N: converte in caratteri minuscoli.

CTRL insieme al tasto R: equivale al CTRL+9.

*A26 COMANDO DI 'TRACE'

Uno dei maggiori problemi che si riscontrano nel COMMODORE 64 e nel VIC 20, durante la stesura e la correzione dei programmi, e' la mancanza del comando TRACE che permette di seguire istruzione dopo istruzione i vari passi del programma. Una mancanza a cui era doveroso sopporre mediante un programma in linguaggio macchina da caricare e conservare in memoria, pronto all'occorrenza ad essere utilizzato.

La parte di memoria prescelta per il suo posizionamento e' quella riservata al caricamento ed al salvataggio dei dati da CASSETTA e quindi sino al momento in cui il REGISTRATORE non viene utilizzato la routine rimarrà in quella zona di memoria.

La routine e' riportata di seguito ed e' da digitare così come e' riportata, pena un non corretto funzionamento.

```
10 FOR X=1 TO 54:READ Y:Z=Z+Y:NEXT
20 IF Z<>6341 THEN PRINT"ERRORE NEI D
ATA":STOP
30 RESTORE
100 PRINT"[CLR]TRACE [GIU']"
110 S=828
120 FORX=S TO S+53:READ Y:POKE X,Y:NE
XT
130 PRINT"SYS ";S+22;" ATTIVA"
140 PRINT"SYS ";S+43;" TOGLIE"
800 DATA 165,58,201,255,240,13
810 DATA 169,60,32,210,255,32
820 DATA 201,189,169,62,32,210
```

```

830 DATA 255,108,251,0,173,8
840 DATA 3,133,251,173,9,3
850 DATA 133,252,169,60,141,8
860 DATA 3,169,3,141,9,3
870 DATA 96,165,251,141,8,3
880 DATA 165,252,141,9,3,96

```

Per utilizzarla dovete battere, come viene anche riportato sul video dopo il lancio del programma che si occupa di caricarla in memoria, il seguente comando dato in modo diretto:

SYS 850

Per disattivarla occorre invece dare il seguente comando, sempre in modo diretto:

SYS 871

Una volta lanciata, essendo il programma in linguaggio macchina ormai caricato in memoria a partire dalla locazione di memoria 828, può essere caricato tranquillamente da DISCO e lanciato il programma BASIC di cui si desidera controllare il funzionamento.

Per quanti non posseggano l'unità a disco, la routine può essere caricata in un'altra zona di memoria, proteggendola dal BASIC, cancellando l'istruzione 110 ed aggiungendo semplicemente la seguente istruzione:

```
115 POKE 56,PEEK(56)-1:CLR:S=5+PEEK(56)*256.
```

Ovviamente dopo tale modifiche non potrà più essere attivata mediante SYS 850 e disattivata mediante SYS 871, ma dovranno essere dati i comandi riportati a video dopo il lancio del programma.

La versione sopra riportata è per il COMMODORE 64, volendo utilizzarla anche sul VIC20, occorre modificare nel seguente modo l'istruzione 20 e l'istruzione 820:

```
20 IF Z<>6373 THEN PRINT"ERRORE NEI DATA":STOP
820 DATA 201,221,169,62,32,210
```

Nel caso si voglia stampare lo svolgersi del programma sotto TRACE dovrà prima essere dato il comando SYS850, poi dovrà essere caricato il programma in memoria su cui si desidera eseguire il TRACE, poi dovrà essere dato il comando OPEN3,4:CMD3 sempre in modo diretto ed infine dovrà essere dato il RUN.

*A27 PAROLA D'ORDINE

La riservatezza è fondamentale nei programmi di tipo GESTIONALE o PROFESSIONALE. Anche nei programmi creati dagli appassionati può però essere interessante evitare che CURIOSI possano mettervi il

naso!... Ecco alcune istruzioni da mettere in testa ai Vostri programmi che fanno al caso nostro:

```

1 POKE 800,100:POKE 819,245
2 PRINT CHR$(147);
3 INPUT "IMMETTERE CODICE ";A
4 IF A <> vostro codice THEN NEW
5 resto del Vostro programma.

```

Ovviamente al posto di: vostro codice va posto un codice numerico come ad esempio: 4567.

*A28 EFFETTO SPECIALE

Per gli amanti degli effetti visivi eccone uno da utilizzare nell'ambito dei Vostri programmi e lungo soltanto un'istruzione. L'effetto è di barre a colori che si muovono velocemente sul bordo del video ed è ottimo per presentazioni e giochi.

```
1 FOR X= 0 TO 1 STEP 0:POKE 53280,3:POKE 53280,6:NEXT
```

*A29 FUNZIONE APPEND

Si rivela molte volte utile la possibilità d'aggiungere dei dati in coda ad un FILE SEQUENZIALE nella gestione dei dati sull'unità a dischi. Ma strano a dirsi un tale comando non viene minimamente menzionato sul manuale dell'unità 1541. Eccolo!... La lettera A posta dopo la S sta' appunto per APPEND.

```
200 OPEN 3,8,3,"0:NOMEFILE,S,A"
```

Al posto di: NOMEFILE, dovrà essere messo il nome del FILE al quale si vogliono appendere i dati.

*A30 ATTESA

A volte si presenta il fatto di far restare il COMPUTER in attesa, ad esempio sino a quando chi sta utilizzando il programma non abbia terminato di leggere dei messaggi presenti sul video, e di far riprendere l'esecuzione del programma mediante la pressione di un tasto. Il programmino sotto riportato provvede appunto a svolgere tale funzione:

```

10 PRINT" PREMERE UN TASTO PER CONTINUARE "
20 POKE 198,0: WAIT 198,1: POKE 198,0

```

*A31 MESSAGGI D'ERRORE

La locazione di memoria 157 controlla la visualizzazione dei messaggi d'errore e di controllo. Si presentano però dei casi in cui si desidera evitare che alcuni oppure tutti tali messaggi non vengano visualizzati.

Ecco alcuni comandi da dare in modo diretto oppure da programma che svolgono questa interessante funzione:

POKE 157,0 disabilita tutti i messaggi .
POKE 157,64 abilita i messaggi d'errore.
POKE 157,128 abilita i messaggi di controllo.
POKE 157,192 abilita sia i messaggi d'errore che i
 messaggi di controllo.

*A32 SIMULAZIONE DELL'ELSE

Il BASIC del COMMODORE 64 e del VIC 20 non possiede la parola chiave ELSE che in molte occasioni si rivela estremamente utile.

Esiste però un trucco che facendo uso delle parole chiave ON e GOTO, riesce in parte a sopperire a tale mancanza, simulando il comportamento della parola chiave ELSE.

Di seguito è riportato un programma esemplificativo di quanto detto che, nell'istruzione 50, utilizza appunto questa tecnica.

```
40 GET A$:IF A$="" THEN 40
50 ON((A$="S")+2)GOTO 60:ON((A$="N")+2)GOTO 70:GOTO 40
60 PRINT"HAI PREMUTO 'S':END
70 PRINT"HAI PREMUTO 'N':END
```



PROGRAMMI

**CM
64**

**DAMA
COMPUTER**

**GG
GG**

Uno dei giochi più conosciuti è certamente quello della DAMA. Pochi non si sono dilettrati almeno una volta con esso. In questa epoca di tecnologia e computer, eccone una versione per computer che Vi farà trascorrere piacevoli momenti.

La versione del gioco implementata sul COMMODORE 64, offre tre tipi di possibilità di gioco: una prima a cui si accede, battendo 0 alla richiesta del numero dei giocatori in cui è il computer che gioca contro se stesso.

Una seconda, a cui si accede battendo 1 alla richiesta del numero dei giocatori ed in cui sarete Voi a giocare contro il computer.

Una terza, a cui si accede battendo 2 alla richiesta del numero dei giocatori, prevista per gestire il gioco di Voi ed un Vostro amico. In questa ultima, il computer funge soltanto da scacchiera.

Per giocare, dovrete comunicare al computer le coordinate di partenza e le coordinate d'arrivo, battendo prima il numero che si riferisce alle righe e subito di seguito il numero che si riferisce alle colonne, e questo per entrambe le coordinate.

Una volta comunicate, il computer visualizzerà la scritta OK?, se la mossa è di Vostro gradimento, dovrete battere la lettera S, quale conferma ed a questo punto la pedina, sarà spostata

sulla scacchiera, secondo le coordinate comunicate.

Nel caso in cui Vi troviate davanti ad una pedina avversaria e desiderate MANGIARLA, dovrete battere la lettera J e dovrete specificare il numero dei salti che volete effettuare. Normalmente, essendo una la pedina che potrete mangiare, dovrete battere il numero 1. Fatto questo, dovrete comunicare le coordinate di partenza e le coordinate d'arrivo, nella stessa identica maniera prima spiegata. Buon divertimento.

SPIEGAZIONE PROGRAMMA

ISTRUZIONI 5-40: inizializzazione variabili, attivazione routine in linguaggio macchina, atto alla creazione della scacchiera sul video; richiesta del numero di giocatori.

ISTRUZIONI 45-64: posizionamento delle pedine dei due giocatori sulla scacchiera.

ISTRUZIONI 70-85: attivazione dei colori utilizzati durante lo svolgimento del gioco.

ISTRUZIONE 90: controllo se è stato battuto il numero 0, alla richiesta del numero dei giocatori e richiamo della parte di programma atta alla sua gestione (istruzioni 400-500).

ISTRUZIONI 95-190: gestione spostamenti pedine sulla scacchiera quando non è richiesto il salto (opzione utilizzate per MANGIARE una pedina avversaria).

ISTRUZIONI 195-210: gestione messaggio OK?.

ISTRUZIONI 215-300: controllo che le coordinate comunicate, siano corrette e che il gioco venga fatto secondo le regole che lo contraddistinguono.

ISTRUZIONI 305-340: parte di programma che si occupa di cancellare le coordinate comunicate e riportate sul video.

ISTRUZIONI 400-500: gestione del gioco, quando a giocare e' il computer.

ISTRUZIONI 600-635: gestione suono finale e vincitore.

ISTRUZIONI 800-845: gestione opzione J (salto) quando e' richiesto un unico salto.

ISTRUZIONI 900-930: gestione opzione J (salto) quando sono richiesti piu' salti.

LISTA VARIABILI DEL PROGRAMMA

PROGRAMMA : DAMACOMPUTER

VAR. : NUMERO ISTRUZIONE

AN#	205	210				
B%	18					
C	400	405	430			
C\$	145	150	155	215	230	920
C2\$	180	185	190	220	230	925
C3\$	830	835	840			
CO	50	55	60			
D	605	615				
DC	230	255	420	425	430	435
DR	230	240	245	250	255	260
	265	270	290	415	420	430
	435	445	455	460	465	480
H	605	610				
I	10	60	65	90	95	230
	260	265	270	285	290	292
	305	335	400	405	435	445
	455	460	475	480	485	490
	625					
J	90	125	195	295	330	800
	900					
J\$	115	120	125			
JF	305	315	900			
JP	10	290	292	480	485	
K	10	225	260	265	275	285
	305	450	455	460	465	475
	490					
L	55	60	215	225	230	270
	280	290	405	445	450	470
	480	605	610	910	915	
L\$	100	105	245	250		
L2	220	235	270	285	290	430
	435	440	445	475	480	

LS	45	55				
M	800	805	810	815	820	825
	830	835	840	845		
N	605	620	625	630		
NK	285	475				
NT	400	410				
P	5	90	335	495		
R	400	405	430	465		
R\$	105	130	135	140	215	230
	920					
R2\$	160	165	170	175	220	230
	275	925				
R3\$	810	815	820			
RQ	30	45	60			
S	600	610	620			
S1	15	18	20			
SN	420	425				
SY	10	60	95	230	270	285
	400	405	435	445	475	625
T	615	620				
Z	18					

1 RESTORE

5 PRINT"[CLR]":INPUT "NUMERO DI GIOCA TORI[3 SP21(0,1 0[2 SP212)";P

10 PRINT"[CLR]":SY(1)=81:SY(2)=102:K=0:I=1:JP(1)=0:JP(2)=0

15 S1=256*PEEK(46)+PEEK(45)+130

18 FORZ=0T0127:READB%:POKE51+Z,B%:NEXT

20 SYSS1

25 PRINT"[HOME]";"[CYN]";"[5 SP21COL 1[3 SP212[3 SP213[3 SP214[3 SP215[3 SP216[3 SP217[3 SP218]";

30 FORRO=1T08:PRINT:PRINT:PRINT"[6 SP 21]";RO:NEXT

35 PRINT"[HOME]":PRINT:PRINT"[5 SP21R ":PRINT"[5 SP211":PRINT"[5 SP218"

40 PRINT"[4 GIU'1[BLU]SALTO[3 GIU'1[4 SIN.1 DA [3 GIU'1[3 SIN.1[GIA]A [GIU '1]";

45 FORRO=1T03:READLS

50 FORCO=1T07STEP2

55 L=LS+4*CO

60 POKEL,SY(I):NEXTCO:NEXTRO

65 I=I+1:IFI<3THEN45

70 POKE55698,6:POKE55699,6:POKE55818, 9:POKE55819,9:POKE55938,7

75 POKE55939,7:POKE55978,7:POKE55979, 7:POKE56018,7:POKE56019,7

80 POKE56058,7:POKE56059,7:POKE56098, 7:POKE56099,7:POKE55578,0

85 POKE56258,2:POKE56259,2:POKE56260,

2

```

90 I=2:J=0:IFP=0THEN400
95 POKE1306,SY(I)
100 GETL$:IFL$=""THEN100
105 IFL$<>"J"THENR$=L$:GOTO135
110 POKE1426,10
115 GETJ$:IFJ$=""THEN115
120 IF((ASC(J$)-53>0)OR(ASC(J$)-49<0)
)THEN115
125 POKE1427,ASC(J$):J=ASC(J$)-48
130 GETR$:IFR$=""THEN130
135 IF((ASC(R$)-56>0)OR(ASC(R$)-49<0)
)THEN100
140 POKE1546,ASC(R$)
145 GETC$:IFC$=""THEN145
150 IF((ASC(C$)-56>0)OR(ASC(C$)-49<0)
)THEN145
155 POKE1547,ASC(C$)
160 GETR2$:IFR2$=""THEN160
165 IFR2$="C"THEN305
170 IF((ASC(R2$)-56>0)OR(ASC(R2$)-49<
0))THEN160
175 POKE1666,ASC(R2$)
180 GETC2$:IFC2$=""THEN180
185 IF((ASC(C2$)-56>0)OR(ASC(C2$)-49<
0))THEN180
190 POKE1667,ASC(C2$)
195 IFJ>1THEN800
200 POKE1986,15:POKE1987,11:POKE1988,
63
205 GETAN$:IFAN$=""THEN205
210 IFAN$<>"S"THEN310
215 L=989+120*(ASC(R$)-48)+4*(ASC(C$)
-48)
220 L2=989+120*(ASC(R2$)-48)+4*(ASC(C
2$)-48)
225 IFPEEK(L+1)<>160THENK=1
230 DR=ASC(R2$)-ASC(R$):DC=ASC(C2$)-A
SC(C$):IFPEEK(L)<>SY(I)THEN310
235 IFPEEK(L2)<>160THEN310
240 IF((ABS(DR)>2)OR(ABS(DR)<1))THEN3
10
245 IF((ABS(DR)=2)AND(L$<>"J"))THEN31
0
250 IF((ABS(DR)=1)AND(L$="J"))THEN310
255 IFABS(DC)<>ABS(DR)THEN310
260 IF((I=1)AND(K=0)AND(DR<0))THEN310
265 IF((I=2)AND(K=0)AND(DR>0))THEN310
270 IF((ABS(DR)=2)AND((PEEK((L+L2)/2)
=SY(I))OR(PEEK((L+L2)/2)=160)))THEN31
0
275 IF((R2$="1")OR(R2$="8"))THENK=1
280 POKEL,160:POKEL+1,160
285 FORNK=0TOK:POKEL2+NK,SY(I):NEXT
290 IFABS(DR)=2THENPOKE(L+L2)/2,160:P
OKE(L+L2)/2+1,160:JP(I)=JP(I)+1
292 IFJP(I)=12THEN600
295 IFJ<2THEN305

```

```

300 GOTO900
305 K=0:JF=0:I=I+1:IFI=3THENI=1
310 POKE1986,32:POKE1987,32:POKE1988,
32
315 IFJF=1THENPOKE1666,160:GOTO160
320 POKE1426,32:POKE1427,32:POKE1546,
32:POKE1547,32:POKE1666,32
325 POKE1667,32:POKE1706,32:POKE1707,
32:POKE1746,32:POKE1747,32
330 POKE1786,32:POKE1787,32:POKE1826,
32:POKE1827,32:J=0
335 IF((P=1)AND(I=1))THEN400
340 GOTO95
400 NT=0:R=INT(8*RND(1)+1):C=INT(8*RN
D(1)+1):POKE1306,SY(I)
405 L=989+120*R+4*C:IFPEEK(L)<>SY(I)T
HEN400
410 NT=NT+1:IFNT>10THEN400
415 DR=INT(3*RND(1)-1):IFDR=0THEN415
420 SN=INT(2*RND(1)):DC=DR
425 IFSN=0THENDC=-DC
430 L2=989+120*(R+DR)+4*(C+DC)
435 IFPEEK(L2)=SY((NOTI)+4)THENL2=L2+
120*DR+4*DC:DR=2*DR:DC=2*DC
440 IFPEEK(L2)<>160THEN410
445 IF((ABS(DR)=2)AND((PEEK((L+L2)/2)
=SY(I))OR(PEEK((L+L2)/2)=160)))THEN41
0
450 IFPEEK(L+1)<>160THENK=1
455 IF((I=1)AND(K=0)AND(DR<0))THEN410
460 IF((I=2)AND(K=0)AND(DR>0))THEN410
465 IF((R+DR=1)OR(R+DR=8))THENK=1
470 POKEL,160:POKEL+1,160
475 FORNK=0TOK:POKEL2+NK,SY(I):NEXT
480 IFABS(DR)=2THENPOKE(L+L2)/2,160:P
OKE(L+L2)/2+1,160:JP(I)=JP(I)+1
485 IFJP(I)=12THEN600
490 K=0:I=I+1:IFI=3THENI=1
495 IFP=0THEN400
500 GOTO95
600 PRINT"[CLR]":S=54272:POKES+24,15:
POKES+5,9:POKES+6,0:POKES3281,0
605 FORN=0TOS:READH,L,D
610 POKES,L:POKES+1,H:POKES+4,33
615 FORT=1TOD:NEXTT
620 POKES+4,32:FORT=1TOD:NEXTT:NEXTN
625 FORN=1TOD:POKE1080+40*N,SY(I):PO
KE1082+40*N,23:POKE1083+40*N,9
630 POKE1084+40*N,14:POKE1085+40*N,19
:NEXTN
635 END
800 FORM=2TOJ
805 POKE1626+40*M,160
810 GETR3$(M):IFR3$(M)=""THEN810
815 IF((ASC(R3$(M))-56>0)OR(ASC(R3$(M)
)-49<0))THEN810
820 POKE1626+40*M,ASC(R3$(M))

```

```

825 POKE1627+40*M,160
830 GETC$(M):IFC$(M)=""THEN830
835 IF((ASC(C$(M))-56>0)OR(ASC(C$(M))
)-49<0))THEN830
840 POKE1627+40*M,ASC(C$(M))
845 NEXTM:GOTO200
900 J=J-1:POKE1427,J+48:JF=1
905 POKE1546,PEEK(1666):POKE1547,PEEK
(1667)
910 FORL=0TO4:POKE1666+L*40,PEEK(1706
+L*40)
915 POKE1667+L*40,PEEK(1707+L*40):NEX
T
920 R$=CHR$(PEEK(1546)):C$=CHR$(PEEK(
1547))
925 R2$=CHR$(PEEK(1666)):C2$=CHR$(PEE
K(1667))
930 GOTO215
1020 DATA169,12,141,32,208,141,33,208
,162,0,142,254,3,142,255,3

```

```

1030 DATA142,252,3,162,48,134,251,134
,253,162,2,142,253,3,162,4
1040 DATA134,252,162,216,134,254,160,
0,169,160,145,251,152,74,74,41
1050 DATA1,170,189,252,3,145,253,200,
192,32,208,236,238,255,3,173
1060 DATA255,3,201,24,208,1,96,238,25
4,3,173,254,3,201,3,208
1070 DATA19,173,252,3,72,173,253,3,14
1,252,3,104,141,253,3,169
1080 DATA0,141,254,3,169,40,24,101,25
1,133,251,133,253,144,183,169
1090 DATA1,24,101,252,133,252,169,1,2
4,101,254,133,254,144,167,96
1100 DATA1109,1233,1349,1713,1829,195
3
1110 DATA16,195,64,21,31,64,25,30,64,
33,135,128,25,30,64,33,135,128
3000 REM -----

```

**CM
64**

UOVA

**GG
GG**

Dovete, utilizzando il joystick, spostare l'occhio riportato sul video, alla ricerca delle uova celate nel campo ed in tal modo, riporle nel cesto evidenziato in basso. Le uova non sono visibili, per cui dovrete procedere per tentativi per riuscire ad individuarle. Quando una d'esse e' stata individuata, si sposta e cade nel cesto, ma attenzione!... Perche' nel medesimo istante tutte le altre, per un istante, si rendono visibili. E' appunto approfittando di questo che potrete facilmente raccoglierle. Il gioco va svolto nel minor tempo possibile.

DESCRIZIONE PROGRAMMA

ISTRUZIONI 110-460: inizializzazione delle variabili del programma e presentazione a video dello stesso.
ISTRUZIONI 470-500: settaggio colori delle SPRITE utilizzate nel programma.
ISTRUZIONI 510-630: attivazione programma e creazione del campo a video.
ISTRUZIONI 640-680: attivazione della SPRITE raffigurante il cesto.
ISTRUZIONI 690-890: attivazione della SPRITE raffigurante l'occhio e sua gestione.
ISTRUZIONI 900-990: loop principale del programma.
ISTRUZIONI 1000-1200: gestione spostamenti dell'occhio nel campo alla ricerca delle uova.
ISTRUZIONI 1210-1220: controllo se e' stato rinvenuto un uovo.
ISTRUZIONI 1230-1460: gestione occhio e gioco per non

ritrovamento dell'uovo.

ISTRUZIONI 1470-1510: gestione nel caso di un ritrovamento di un uovo.

ISTRUZIONI 1520-1720: lampeggio delle uova rimaste.

ISTRUZIONI 1730-1770: riinizializzazione variabili programma inerenti ai puntatori delle uova rimaste.

ISTRUZIONI 1780-2060: gestione della parte finale del gioco e dei relativi messaggi ad essa inerenti.

ISTRUZIONI 2250-2270: inizializzazione del sintetizzatore sonoro del COMMODORE 64.

ISTRUZIONI 2280-2310: gestione del suono in caso di ritrovamento di un uovo.

LISTA VARIABILI DEL PROGRAMMA

PROGRAMMA : UOVA

VAR. : NUMERO ISTRUZIONE

BC\$	:	270	570				
BK\$	:	190	910				
BL\$	:	180	680				
CC\$	:	330	340	350	650	1820	1930
CK	:	1220	1230	1240			
CL\$	:	160	350	510	2050		
DF	:	130	870	1540			
EC	:	290	850				
FB	:	960	970				
FF	:	970	1010	1020	1030	1040	1050

```

      : 1060 1070 1080
GR$ : 170 370
HO$ : 160 330 520 550 680 880
      : 910 920 1800 1820
HS : 240 1940 1970
IN : 240 1010 1020 1030 1040 1050
      : 1060 1070 1080
J : 940 950 960 980 2010 2020
J2 : 210 2010
KT : 1540 1550
PK : 1240 1250 1270 1320 1390 1420
      : 1470 1550
PU$ : 190
Q : 1740 1750
R : 560 570
R( : 230 240 560 740 770 850
RE$ : 170 380 520 920 1820
RO$ : 200 520 1870
RV$ : 200 520 570 660 680 1840
      : 1860 1890 1900
S : 1240 1250 1280 1300 1310 1390
      : 1410 1460
S( : 310 800 810 1250
SC : 920 930 1480 1790 1940 1960
      : 2030
SD : 220 1380 1450 1490 2260 2270
      : 2290 2300 2310
SX : 1310 1320 1330 1340 1360
SY : 1300 1440
T1 : 1290 1480
T2 : 1290 1480
T3 : 1290 1480
TI : 240 910 1790
TI$ : 890
V : 210 430 480 490 500 540
      : 640 670 690 700 720 740
      : 790 800 810 830 840 860
      : 870 1130 1150 1170 1180 1190
      : 1200 1220 1300 1310 1320 1390
      : 1410 1420 1460 1470 1500 1540
      : 1550 1790 2050
X : 230 240 260 290 310 330
      : 420 430 460 560 590 610
      : 730 740 760 790 800 810
      : 840 850 870 1240 1250 1360
      : 1370 1380 1390 1410 1430 1440
      : 1450 1460 1530 1770 1860 1890
      : 2060
XE : 710 1030 1040 1050 1060 1070
      : 1080 1110 1120 1140 1150 1160
      : 1170 1180 1190 1200
XP : 770 780 790 810
Y : 420 430 1880
YE : 710 1010 1020 1040 1050 1070
      : 1080 1090 1100 1130
YE$ : 180 360 660
Z : 260 270 1330 1360 2060

```

```

110 REM
120 REM
130 DF=7:REM ---
140 REM ---
150 POKES3280,7:POKE53281,1
160 CL$=CHR$(147):HO$=CHR$(19):PRINTC
L$
170 RE$=CHR$(150):GR$=CHR$(153)
180 YE$=CHR$(158):BL$=CHR$(154)
190 PU$=CHR$(156):BK$=CHR$(144)
200 RV$=CHR$(18):RO$=CHR$(146)
210 J2=56320:V=53248:POKEV+21,0
220 SD=54272
230 DEFFNR(X)=INT(RND(1)*X)+1
240 X=FNR(-TI):IN=4:HS=99999
250 POKES3272,21
260 FORX=1TO10:READZ
270 BC$=BC$+CHR$(Z):NEXT
280 DATA 5,5,5,5,30,30,153,153,158,15
5
290 FORX=1TO6:READC(X):NEXT
300 DATA 3,4,8,10,14,14
310 FOR X=0TO7:READS(X):NEXT
320 DATA 1,2,4,8,16,32,64,128
330 CC$=HO$:FORX=1TO25
340 CC$=CC$+CHR$(17):NEXT
350 PRINTCL$LEFT$(CC$,12);
360 PRINTYE$TAB(12)"RACCOGLI UOVA !!"
370 PRINTGR$TAB(12)"-----"
380 PRINT:PRINTRE$
390 PRINTTAB(9)"CERCA LE CINQUE UOVA
"
400 PRINT
410 PRINTTAB(10)"IL PIU' VELOCEMENTE"
420 FORX=832TO1022:READY
430 POKEV,Y:NEXTX:POKEV+16,0
440 GOSUB2060
450 POKE2040,15:POKE2041,14
460 FORX=2042TO2047:POKEV,13:NEXT
470 REM ---
480 POKEV+39,6:REM ---
490 POKEV+40,11:REM[2 SP2]---
500 POKEV+41,1:REM ---
510 PRINTCL$
520 PRINTHO$RV$RE$"PRONTO.."RO$;
530 PRINT"[11 SP2]"
540 POKEV+21,0
550 PRINTHO$
560 FORX=41TO840:R=FNR(10)
570 PRINTMID$(BC$,R,1)RV$" ";
580 NEXT:PRINT
590 PRINTCHR$(30):FORX=1TO40
600 PRINTCHR$(168);NEXT
610 FORX=1TO39:PRINTCHR$(166);NEXT
620 POKE2023,102:POKE56295,5
630 GOSUB1730
640 POKEV+2,96:POKEV+3,221:REM ---

```

```

650 PRINT LEFT$(CC$,24);
660 PRINTTAB(9)RV$YE$"[3 SPZ]";
670 POKEV+21,2
680 PRINTH$RV$BL$"ATTIVAZIONE"
690 POKEV,200:POKEV+1,215:REM ---
700 POKEV+4,204:POKEV+5,222:REM ---
710 YE=215:XE=200
720 POKEV+21,255
730 FORX=7T015STEP2
740 POKEV+X,FNR(145)+60
750 NEXT
760 FORX=3T07
770 XP=FNR(250)+25
780 IFXP<255THEN810
790 POKEV+X*2,XP-255
800 POKEV+16,(PEEK(V+16)ORS(X)):GOTO8
20
810 POKEV+X*2,XP:POKEV+16,PEEK(V+16)A
ND255-S(X)
820 NEXT
830 IFPEEK(V+30)<>5THEN730
840 FORX=V+42TOV+46
850 POKEV,EC(FNR(6)):NEXT
860 POKEV+21,255
870 FORX=1T0240-20*DF:POKEV+27,0:POKE
V+27,248:NEXT
880 PRINT H0$"[19 SPZ]"
890 T1$="000000":GOSUB2280
900 REM ---
910 PRINTH$BK$TAB(33)T1
920 PRINTH$RE$"UOVA: "SC
930 IF SC=5THEN1780
940 J=31-(PEEK(56320)AND31)
950 IFJ=0THEN900
960 FB=JAND16:IFFBTHENJ=ABS(16-J)
970 FF=1:IFFBTHENFF=4
980 ONJGOSUB1010,1020,1000,1030,1040,
1050,1000,1060,1070,1080
990 GOTO1090
1000 RETURN:REM ---
1010 YE=YE-IN*FF:RETURN
1020 YE=YE+IN*FF:RETURN
1030 XE=XE-IN*FF:RETURN
1040 XE=XE-IN*FF:YE=YE-IN*FF:RETURN
1050 XE=XE-IN*FF:YE=YE+IN*FF:RETURN
1060 XE=XE+IN*FF:RETURN
1070 XE=XE+IN*FF:YE=YE-IN*FF:RETURN
1080 XE=XE+IN*FF:YE=YE+IN*FF:RETURN
1090 IFYE<52THENYE=52
1100 IFYE>204THENYE=204
1110 IFXE>322THENXE=322
1120 IFXE<20THENXE=20
1130 POKEV+1,YE:POKEV+5,YE+7
1140 IFXE>255THEN1190
1150 POKEV,XE:POKEV+16,PEEK(V+16)AND2
54
1160 IFXE+4>255THEN1180

```

```

1170 POKEV+4,XE+4:POKEV+16,PEEK(V+16)
AND251:GOTO1210
1180 POKEV+4,XE+4-255:POKEV+16,PEEK(V
+16)OR4:GOTO1210
1190 POKEV,XE-255:POKEV+16,PEEK(V+16)
OR1
1200 POKEV+4,XE+4-255:POKEV+16,PEEK(V
+16)OR4:GOTO1210
1210 REM ----
1220 CK=PEEK(V+30):POKEV+30,0
1230 IFCK=5ORCK=0THEN900:REM ---
1240 S=0:PK=CK-5:FORX=3T07
1250 IFPK=S(X)THENS=X
1260 NEXT
1270 IFPKAND1THEN900
1280 IFS=0THEN900
1290 T3=PEEK(162):T2=PEEK(161):T1=PEE
K(160)
1300 SY=PEEK(V+1+S*2)
1310 SX=PEEK(V+S*2)
1320 IFPEEK(V+16)ANDPK=PKTHENSX=SX+25
4
1330 Z=1:IFSX>100THENZ=-1
1340 IFSX>300THENSX=SX-255
1350 GOSUB2250
1360 FORX=SXT0100STEPZ*5:GOSUB1520
1370 IFX>255THEN1410
1380 POKESD+1,255-X:POKESD+4,17
1390 POKEV+S*2,X:POKEV+16,PEEK(V+16)A
ND255-PK
1400 GOTO1430
1410 POKEV+S*2,X-255
1420 POKEV+16,PEEK(V+16)ORPK
1430 NEXTX
1440 FORX=SYT0230STEP3:GOSUB1520
1450 POKESD+1,255-X:POKESD+4,17
1460 POKEV+1+S*2,X:NEXT
1470 POKEV+21,PEEK(V+21)AND255-PK:REM
-----
1480 SC=SC+1:POKE162,T3:POKE161,T2:PO
KE160,T1
1490 POKESD+4,0
1500 POKEV+27,248
1510 GOTO900
1520 REM -----
1530 IFX/4<>INT(X/4)THEN RETURN
1540 IFKT=DFTHENKT=1:POKEV+27,0:RETUR
N
1550 KT=KT+1:POKEV+27,248-PK:RETURN
1560 DATA 7,224,0,31,248,0,63,252,0,6
3
1570 DATA 252,0,31,248,0,7,224,0,0,0,
0
1580 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0
1590 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0
1600 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0
,0

```

```

1610 DATA 0,254,0,3,1,128,12,0,96,16
1620 DATA 0,16,32,0,8,64,0,4,64,0,4
1630 DATA 128,0,2,128,0,2,128,0,2,128
1640 DATA 0,2,96,0,12,223,255,246,128
1650 DATA 0,2,152,7,130,135,224,26,17
1660 DATA 15,130,135,128,26,128,63,2
1670 DATA 112,0,12,31,255,240,0,0,0,0
1680 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0
1690 DATA 0,255,0,3,0,192,4,60,32,8
1700 DATA 118,16,8,126,16,4,60,32,3,0
1710 DATA 192,0,255,0,0,0,0,0,0,0,0
1720 DATA 0,0,0,0,0,0,0,0,0,0,0,0
1730 REM -----
1740 FORQ=217TO242
1750 IFPEEK(Q)<128THENPOKEQ,PEEK(Q)+1
1760 NEXT:RETURN
1770 FORX=1TO28:PRINTCHR$(192);:NEXT:
RETURN
1780 REM -----
1790 SC=TI:POKEV+21,2
1800 PRINHTO$TAB(33)"[6 SPZ]"
1810 SYS49152:POKE53280,15:POKE53281,
1
1820 PRINHTO$LEFT$(CC$,5)RE$
1830 GOSUB 1730
1840 PRINTTAB(5)RV$CHR$(176);:GOSUB17
70
1850 PRINTCHR$(174)
1860 FORX=1TO9:PRINTTAB(5)RV$CHR$(221
);
1870 PRINTRO$;
1880 FORY=1TO28:PRINT" ";:NEXTY
1890 PRINTRV$CHR$(221):NEXTX
1900 PRINTTAB(5)RV$CHR$(173);:GOSUB17
70
1910 PRINTCHR$(189)
1920 GOSUB1730
1930 PRINTCHR$(149)LEFT$(CC$,7)
1940 IFSC<=HSTHENHS=SC

```

```

1950 PRINTTAB(11)"FINE GIOCO":PRINT
1960 PRINTTAB(8)"[5 SPZ]PUNTI="SC
1970 PRINTTAB(8)"MASS.PUNTI="HS
1980 PRINT
1990 PRINTTAB(11)"PREMI PULS. DI FUOC
0 "
2000 PRINTTAB(11)"ANCORA[4 SPZ]"
2010 WAIT J2,127,127:J=PEEK(J2)
2020 IFJAND16THEN2050
2030 SC=0:SYS49214
2040 POKE53280,7:POKE53281,1:GOTO640
2050 POKEV+21,0:PRINTCL$:END
2060 FORX=49152TO49289:READZ:POKEX,Z:
NEXT:RETURN
2070 DATA 32,124,192,162,0,189,0,4
2080 DATA 157,0,193,189,0,5,157,0
2090 DATA 194,189,0,6,157,0,195,189
2100 DATA 0,216,157,0,197,189,0,217
2110 DATA 157,0,198,189,0,218,157,0
2120 DATA 199,202,208,217,162,216,189
2130 DATA 0,7,157,0,196,89,0,219,157
2140 DATA 0,200,202,208,241,96,32,124
2150 DATA 192,162,0,189,0,193,157,0
2160 DATA 4,189,0,194,157,0,5,189
2170 DATA 0,195,157,0,6,189,0,197
2180 DATA 157,0,216,189,0,198,157,0
2190 DATA 217,189,0,199,157,0,218,202
2200 DATA 208,217,162,216,189,0,196,1
57
2210 DATA 0,7,189,0,200,157,0,219,202
2220 DATA 208,241,96,162,0,181,217
2230 DATA 9,128,149,217,232,224,25
2240 DATA 208,245,96
2250 REM -----
2260 POKESD+24,15:POKESD+5,192
2270 POKESD+6,250:POKESD+4,0:RETURN
2280 REM -----
2290 POKESD+24,15:POKESD+5,9:POKESD+1
,20:POKESD+4,128
2300 POKESD+6,0
2310 POKESD+4,129:RETURN

```

**CM
64**

**DOPPIA
COPPIA**

**GG
GG**

Se non avete mai provato il gioco delle coppie, questo programma Ve ne offre la possibilità'.

Il gioco consiste nello scegliere due carte da un mazzo di 54, sperando di prenderne due di tipo uguale, come ad esempio due assi oppure due donne. Nel caso azzeccate una coppia di carte di ugual tipo, potrete continuare sempre a giocare, sino a quando non

sbaglierete. In tal caso il gioco passerebbe al Vostro rivale. Vince chi ovviamente prende piu' coppie di carte di uguale tipo. Buon divertimento.

DESCRIZIONE PROGRAMMA

ISTRUZIONI 100-135: inizializzazione variabili del programma.

ISTRUZIONI 140-150: gestione tipo di gioco: contro il computer o contro un avversario umano.

ISTRUZIONI 160-180: lettura e caricamento valori dai DATA presenti nel programma.

ISTRUZIONI 190-250: gestione della simulazione del rimiscolamento del mazzo di carte e della stesura delle stesse sul tavolo da gioco a faccia coperta.

ISTRUZIONI 260-410: gestione del gioco dei relativi giocatori e degli eventuali messaggi inerenti al gioco.

ISTRUZIONI 415-490: gestione della parte conclusiva del gioco e dei relativi messaggi ad essa inerenti.

ISTRUZIONI 500-550: routine che si occupa di togliere dal video le eventuali carte dello stesso tipo trovate dai giocatori.

ISTRUZIONI 560-760: gestione del gioco, quando chi a giocare e' il computer.

LISTA VARIABILI DEL PROGRAMMA

PROGRAMMA : DOPPIA COPPIA

VAR. : NUMERO ISTRUZIONE

AN\$	470	480				
BB\$	160					
BL	170	510				
CD	112	175	230			
CL	210	220	230			
CM	440	450	455	460	560	640
CO	290	291	295	300	305	315
	316	320	325	330	335	350
	355	500	510	570	575	580
	590	600	610	615	620	625
	640	645	670	680	700	710
	730	740	750			
EL	510	525	535			
FL	400	405	415	565		
GL\$	160	170	180			
HS	400	415	420			
I	115	120	160	170	175	180
	190	230	250	260	280	350
	360	400	405	410	430	435
J	160	180	200	230	250	270
	360	505	510	525	535	540
	560	650	690	700	710	720
K	360	525	535	650	690	700
	710	720				
NL\$	135	270	560			
NM\$	160	180				
NP	141	150	260	370	390	400
	430					
PL	135	190	230	240		
PM	280	350	400	405	415	430
	440	450	455	460	565	
PMB	405					

RO	290	291	295	310	315	316
	320	325	340	350	355	500
	510	570	575	580	590	600
	630	640	645	670	680	700
	710	730	740	750		
RW	210	220	230			
S	110	520	525	530	535	
SD	112	230	295	325	350	355
	500	510	575	600	640	645
	680	710	740			
ST	170	230				
TM	545					
UD	112	220	230	355	500	645
	710					
V	110	130	300	305	310	330
	335	340	360	540	610	615
	620	625	630	650		
X	112	175	300	305	330	335
	610	615	620	625		
Y	170	310	340	630		

```

100 CLR:RESTORE
105 PRINT"[CLR][5 GIU'][SHFT {]}TAB(1
8)"DOPPIA COPPIA"TAB(48)"*****
*****"
110 S=54272:V=53248:POKEV+33,6:POKEV+
32,14
112 DIMX(13),CD(13),UD(4,13),SD(4,13)
115 FORI=0TO2:POKE12288+I,0:NEXTI
120 FORI=3TO62STEP3:POKE12288+I,15:PO
KE12289+I,255:POKE12290+I,240:NEXTI
130 POKE2040,192:POKE2041,192:POKEV+2
1,3:POKEV+27,3:POKEV+39,1:POKEV+40,1
135 READNL$:PL=1264
140 PRINT"[GIU']CIOCO CONTRO IL COMPU
TER BATTI 1[8 SPZ]"
141 INPUT"GIOCO CON ALTRE PERSONE MAX
(2-10)";NP
150 IFNP<10RNP>10THEN140
160 FORI=1TO3:READBB$:FORJ=1TO13:GL$(
I)=GL$(I)+BB$:NEXTJ:NEXTI:READNM$
170 GL$(4)=GL$(3):GL$(3)=GL$(2):FORI=
1TO4:READY(I),ST(I),BL(I):NEXTI
175 FORI=1TO13:READX(I),CD(I):NEXTI
180 PRINT"[CLR][10 GIU']":PRINTNM$:FO
RI=1TO4:FORJ=1TO4:PRINTGL$(J):NEXT:NE
XT
190 FORI=1TO4:PL=PL+121
200 FORJ=1TO13
210 RW=INT(RND(1)*4+1):CL=INT(RND(1)*
13+1)
220 IFUD(RW,CL)=32THEN210
230 SD(I,J)=CD(CL):UD(RW,CL)=32:POKEP
L,CD(CL):POKEPL+40,ST(RW)

```

```

240 POKEPL+54272,6:POKEPL+54312,6:PL=
PL+3
250 NEXTJ:NEXTI
260 FORI=1TONP
270 PRINT"[HOME]":FORJ=1TO6:PRINTNL$:
NEXTJ
280 PRINT"[HOME][GIU']GICATORE NUM. "
;I,"MANI[4 SPZ]";PM(I)
290 INPUT"SCEGLI LA PRIMA CARTA(RI.,C
O.)";RO(1),CO(1)
291 IFRO(1)<1ORRO(1)>4ORCO(1)<1ORCO(1
)>13THEN280
295 IFSD(RO(1),CO(1))=32THENPRINT"PAS
SATA !!!":GOTO280
300 IFX(CO(1))>255THENPOKEV,X(CO(1))-
256:POKEV+16,PEEK(V+16)OR1:GOTO310
305 POKEV,X(CO(1))
310 POKEV+1,Y(RO(1))
315 INPUT"SECONDA CARTA PREGO[2 SPZ](
RIGA,COL)";RO(2),CO(2)
316 IFRO(2)<1ORRO(2)>4ORCO(2)<1ORCO(2
)>13THEN315
320 IFRO(1)=RO(2)ANDCO(1)=CO(2)THENPR
INT"STESSA CARTA NON VALIDA":GOTO280
325 IFSD(RO(2),CO(2))=32THENPRINT"PAS
SATA!!!":GOTO315
330 IFX(CO(2))>255THENPOKEV+2,X(CO(2)
)-256:POKEV+16,PEEK(V+16)OR2:GOTO340
335 POKEV+2,X(CO(2))
340 POKEV+3,Y(RO(2))
350 IFSD(RO(1),CO(1))=SD(RO(2),CO(2))
THENPM(I)=PM(I)+1:GOSUB500:GOTO270
355 UD(RO(1),CO(1))=SD(RO(1),CO(1)):U
D(RO(2),CO(2))=SD(RO(2),CO(2))
360 FORJ=1TO2700:NEXTJ:FORK=0TO3:POKE
V+K,0:NEXTK:POKEV+16,0:NEXTI
370 IFNP=1THENGOSUB560
380 GOTO260
390 IFNP=1THEN440
400 HS=1:FORI=2TONP:IFPM(I)>PM(HS)THE
NHS=I:FL=0:GOTO410
405 IFPM(I)=PM(HS)THENFL=1
410 NEXTI
415 IFFL=1THENPRINT"[CLR][GIU']POSSIE
DE[12 SPZ]";PM(HS);" MANI[3 SPZ]":GOT
O430
420 PRINT"[CLR][3 GIU']JHA VINTO IL GI
OCATORE[4 SPZ]#";HS
430 PRINT:PRINT:FORI=1TONP:PRINT"GIOC
ATORE#";I,PM(I);" MANI[3 SPZ]"
435 NEXTI:GOTO470
440 IFCM>PM(1)THENPRINT"[CLR][3 GIU']
HO VINTO! HO VINTO !"
450 IFCM<PM(1)THENPRINT"[HOME][3 GIU'
JACC!! HAI VINTO"
455 IFPM(1)=CMTHENPRINT"[CLR][3 GIU']
PARI!!!!!!!"

```

```

460 PRINT"[2 GIU']JHAI FATTO";PM(1);"
MANI[3 SPZ]"TAB(40)"HO FATTO[3 SPZ]";
CM;" MANI[3 SPZ]"
470 INPUT"[GIU']JUN ALTRO GIOCO[10 SPZ
]";AN$
480 IFAN$="SI"THENGOTO100
490 END
500 UD(RO(1),CO(1))=32:UD(RO(2),CO(2)
)=32:SD(RO(1),CO(1))=32
505 FORJ=1TO1500:NEXTJ
510 SD(RO(2),CO(2))=32:FORJ=1TO2:EL(J
)=BL(RO(J))+3*CO(J):NEXTJ
520 POKES+24,15:POKES+5,65:POKES+6,13
6:POKES+1,57:POKES,172:POKES+4,17
525 FORJ=0TO120STEP40:FORK=0TO2:POKEE
L(1)+J+K,32:NEXTK:NEXTJ:POKES+4,16
530 POKES+1,40:POKES,200:POKES+4,17
535 FORJ=0TO120STEP40:FORK=0TO2:POKEE
L(2)+J+K,32:NEXTK:NEXTJ:POKES+4,16
540 FORJ=0TO3:POKEV+J,0:NEXTJ:POKEV+1
6,0
545 TM=TM+1:IFTM=26THEN390
550 RETURN
560 PRINT"[HOME]":FORJ=1TO6:PRINTNL$:
NEXTJ:PRINT"[HOME][GIU']GIOCO IO",CM;
" MANI[3 SPZ]"
565 FL=FL+1:IFFL/2=INT(FL/2)ORPM(1)>1
1THENGOTO670
570 RO(1)=INT(RND(1)*4+1):CO(1)=INT(R
ND(1)*13+1)
575 IFSD(RO(1),CO(1))=32THEN570
580 RO(2)=INT(RND(1)*4+1):CO(2)=INT(R
ND(1)*13+1)
590 IFRO(1)=RO(2)ANDCO(1)=CO(2)THEN58
0
600 IFSD(RO(2),CO(2))=32THEN580
610 IFX(CO(1))>255THENPOKEV,X(CO(1))-
256:POKEV+16,PEEK(V+16)OR1:GOTO620
615 POKEV,X(CO(1))
620 IFX(CO(2))>255THENPOKEV+2,X(CO(2)
)-256:POKEV+16,PEEK(V+16)OR2:GOTO630
625 POKEV+2,X(CO(2))
630 POKEV+1,Y(RO(1)):POKEV+3,Y(RO(2))
640 IFSD(RO(1),CO(1))=SD(RO(2),CO(2))
THENCM=CM+1:GOSUB500:GOTO560
645 UD(RO(1),CO(1))=SD(RO(1),CO(1)):U
D(RO(2),CO(2))=SD(RO(2),CO(2))
650 FORJ=1TO2700:NEXTJ:FORK=0TO3:POKE
V+K,0:NEXTK:POKEV+16,0
660 RETURN
670 RO(1)=INT(RND(1)*4+1):CO(1)=INT(R
ND(1)*13+1)
680 IFSD(RO(1),CO(1))=32THEN670
690 FORJ=1TO4:FORK=1TO13
700 IFRO(1)=JANDCO(1)=KTHEN720
710 IFSD(RO(1),CO(1))=UD(J,K)THENRO(2
)=J:CO(2)=K:GOTO610

```

```

720 NEXTK:NEXTJ
730 RO(2)=INT(RND(1)*4+1):CO(2)=INT(R
ND(1)*13+1)
740 IFSD(RO(2),CO(2))=32THEN730
750 IFRO(1)=RO(2)ANDCO(1)=CO(2)THEN73
0
760 GOTO610
770 DATA"[38 SPZ]", "[SHFT U][SHFT C][
SHFT I]", "[SHFT B][SHFT B]", "[SHFT J
][SHFT C][SHFT K]"

```

```

775 DATA" 1[2 SPZ]2[2 SPZ]3[2 SPZ]4[2
SPZ]5[2 SPZ]6[2 SPZ]7[2 SPZ]8[2 SPZ]
9[2 SPZ]10 11 12 13"
780 DATA 119,65,1341,151,83,1501,183,
88,1661,215,90,1821
790 DATA 24,1,48,11,72,17,96,10,120,2
0,144,57,168,56,192,55
800 DATA 216,54,240,53,264,52,288,51,
312,50

```

CM DUMP UU 64 GRAFICO UU

Un interessante ausilio per trasportare mediante stampante, su carta quanto in quel momento e' presente sul video del COMMODORE 64. Questo si puo' ottenere semplicemente digitando il programma sotto riportato, lanciandolo e premendo il tasto di funzione F1. Al lancio del programma Vi verranno presentate due possibili scelte: caratteri minuscoli oppure caratteri maiuscoli. Premendo il tasto contrassegnato dalla lettera L, otterrete le minuscole, mentre premendo il tasto contrassegnato dalla lettera U, otterrete le maiuscole.

La solita raccomandazione: salvate sempre il programma, prima di effettuarne qualsiasi lancio!... In caso contrario correreste il rischio, nell'eventualita' di aver commesso qualche errore di digitazione, di perderlo.

DESCRIZIONE PROGRAMMA

ISTRUZIONI 10-240: sono contenuti i valori numerici da caricare in memoria corrispondenti al programma in linguaggio macchina, incaricato di svolgere la funzione di DUMP od HARD-COPY del video sulla stampante.

ISTRUZIONI 250-320: lettura dei valori contenuti nelle istruzioni prima viste e loro caricamento in memoria a partire dalla locazione 49400 fino alla locazione 49152.

ISTRUZIONI 330-340: gestione richiesta di stampa a caratteri minuscoli o maiuscoli.

ISTRUZIONE 600: istruzione alla quale si perviene mediante l'istruzione 310, nel caso di valori errati presenti dopo la parola chiave DATA, nelle istruzioni che vanno dalla 10 alla 240.

LISTA VARIABILI DEL PROGRAMMA

PROGRAMMA : DUMP GRAFICO

VAR. : NUMERO ISTRUZIONE

A\$	:	330			
CS	:	270	280	310	
CS1	:	250	260	310	
EA	:	250	270	290	
I	:	260	270	280	290 320
J	:	270	280	290	300
K	:	280			
L	:	250	310	320	600
SA	:	250	270		

```

10 DATA 1136,1059,1515,1764,1699,1690
,1762
20 DATA 1823,1612,1993,1902,1747,1366
,1640
30 DATA 1945,1578,1724,1493,1472,1865
,1213
40 DATA 120,173,20,3,141,169,2,173,21
,3,141,170
50 DATA 2,169,29,141,20,3,169,192,141
,21,3,169
60 DATA 0,133,254,88,96,165,197,201,4
,208,4,165
70 DATA 254,240,3,108,169,2,169,255,1
33,254,169,8
80 DATA 162,4,160,255,32,186,255,169,
0,32,189,255
90 DATA 32,192,255,162,8,32,201,255,1
69,0,133,251
100 DATA 133,253,169,4,133,252,162,25
,32,230,192,177
110 DATA 251,32,139,192,200,204,168,2
,208,245,169,13
120 DATA 32,210,255,169,40,24,101,251
,133,251,144,2
130 DATA 230,252,32,230,192,202,208,2
23,169,13,32,210
140 DATA 255,32,210,255,32,204,255,16
9,8,32,195,255

```

```

150 DATA 169,0,133,254,76,188,254,142
,167,2,170,192
160 DATA 0,208,15,169,16,32,210,255,1
69,50,32,210
170 DATA 255,169,48,32,210,255,138,20
1,128,144,19,41
180 DATA 127,170,165,253,208,25,169,2
55,133,253,169,18
190 DATA 32,210,255,56,176,13,165,253
,240,9,169,0
200 DATA 133,253,169,146,32,210,255,1
38,201,32,144,11
210 DATA 201,64,176,11,32,210,255,174
,167,2,96,105
220 DATA 64,144,245,201,96,176,4,105,
32,144,237,24
230 DATA 144,241,160,39,177,251,136,2

```

```

40,4,201,32,240
240 DATA 247,200,200,140,168,2,160,0,
96
250 DIMCS(21):SA=49152:EA=49400:L=1
260 FORI=1TO21:READCS(I):NEXT
270 FORI=SATOEASTEP12:CS=0:FORJ=0TO11
280 READK:POKEI+J,K:CS=CS+K
290 IFI+J=EATHENJ=11:I=EA
300 NEXTJ
310 IFCS<>CS(L)THEN600
320 L=L+1:NEXTI
330 PRINT"[GIU']MAIUSCOLE O MINUSC.";
:INPUT"[3 SPZ]U[3 SIN.]"A$:IFA$="L"
HENPOKE49203,7
340 SYS49152:NEW
600 PRINT"ERRORE IN DATA ALL'ISTR.#";
40+(L-1)*10

```

**VC
20**

RANE

**GG
GG**

Vi trovate nei panni di una rana e dovete cercare d'attraversare una strada densa di traffico senza essere investiti dai veicoli che sfrecciano veloci. Nel caso riusciate ad attraversarla incolumi, non pensate di aver terminato, perche' Vi aspetta un insidioso canale da attraversare, saltando da un tronco al successivo sino a raggiungere finalmente la Vostra tana posta in alto ed incrementare in tal modo il Vostro punteggio.

DESCRIZIONE PROGRAMMA

ISTRUZIONI 5-50: inizializzazioni variabili e richiamo della subroutine posta dall'istruzione 650 all'istruzione 700 che si occupa della modifica del set standard dei caratteri e di caricare in memoria i valori numerici della parte in linguaggio macchina.

ISTRUZIONI 60-120: creazione dello schermo video inerente al gioco.

ISTRUZIONI 130-460: ciclo principale del programma che si occupa di muovere i veicoli, le rane, i tronchi e di rilevare le eventuali collisioni.

ISTRUZIONI 470-510: gestione della parte finale del gioco con i relativi messaggi di conmiato.

LISTA VARIABILI DEL PROGRAMMA

PROGRAMMA : RANE

VAR. : NUMERO ISTRUZIONE

A	:	140	150	160	170				
A\$	:	490	500						
DX	:	140	160	170	190	210	260		
	:	280	290	430	441	530			
DY	:	140	150	200	210	240	250		
	:	260	280	290	430	441	530		
F1	:	60	130	560	590				
F2	:	60	130	570	590				
F3	:	60	130	580	590				
F4	:	60	130	585	590				
FT	:	60	590						
G	:	540	700						
HI	:	370	390						
K	:	655	700						
L	:	60	380	450					
OC	:	70	270	280	420	520			
OP	:	70	270	280	420	520			
P	:	210	220	230	240	250	350		
S1	:	60	140	150	160	170	180		
	:	441	540						
SC	:	60	280	370	390	550	590		
T	:	441	460	540	550	655	695		
TX	:	60	400	590	595				
TI	:	400							
TI\$	:	70	380						
X	:	70	190	210	270	280	290		
	:	300	310	320	350	360	420		
	:	430	441	520	530	560	570		
	:	580	585	655	810				
Y	:	70	200	210	220	230	240		
	:	250	270	280	290	300	310		
	:	320	350	360	420	430	441		
	:	520	530						

```

5 POKE37159,255:POKE36878,15:POKE37879
,255
10 POKE36879,8:PRINT"[CLR][VRD][3 GIU'
][RVS ON][8 SPZ][RANA][10 SPZ][RVS OFF]"
20 PRINT"[2 GIU'] [BNC]Z-[ROSS]SIN. [B
NC]X-[ROSS]DEST. [BNC]F1-[ROSS]SU":PRI
NT"[GIU'] [7 SPZ][BNC]F3-[ROSS]GIU'"
30 PRINT"[CYN][3 GIU'] [6 SPZ]*****
*"
40 PRINT"[2 GIU'] [BNC] [7 SPZ]PRONTO!"
50 GOSUB650:CLR
60 SC=0:L=3:S1=36875:F1=32:F2=32:F3=32
:F4=32:TX=600:FT=0:POKE36879,12
70 OP=32:OC=1:Y=10:X=10:TI$="000000":P
OKE36878,15
80 PRINT"[CLR][CYN]''''''''''''''''
''':PRINT"[ROSS]'''''' [2 SPZ]'' [4 SPZ
]''''
90 PRINT"[GIU'] [3 SPZ]'''' [6 SPZ]''
''':PRINT"[GIU'] [2 SPZ]'''' [8 SPZ]''
''
100 PRINT"[GIU'] [2 SPZ]'' [4 SPZ]'' [4
SPZ]''':PRINT"[GIU'] [CYN]''''''''''
''''''''''
110 PRINT"[CYN] &() [3 SPZ][PORP] &() [2
SPZ][ROSS] [2 SPZ]&()":PRINT"[GIU'] [RO
SS] % [PORP] % [2 SPZ][VRD] % [2 SPZ][
GIA] %"
120 PRINT"[GIU'] [VRD] [2 SPZ]&() [4 SPZ]
[CYN] &() [2 SPZ][BNC] &()":PRINT"[GIU'
][PORP] [2 SPZ] % [3 SPZ][BNC] [2 SPZ] % [
3 SPZ][GIA] % [2 GIU'] [VRD]"
130 POKE7683,F1:POKE7688,F2:POKE7693,F
3:POKE7698,F4
131 POKE38403,5:POKE38408,5:POKE38413,
5:POKE38418,5
140 DY=0:DX=0:A=PEEK(197):IFA=39THENDY
=-1:POKE$1,200
150 IFA=47THENDY=1:POKE$1,210
160 IFA=33THENDX=-1:POKE$1,220
170 IFA=26THENDX=1:POKE$1,230
180 POKE$1,0
190 IFX+DX<10RX+DX>20THEN420
200 IFY+DY>10THEN300
210 P=PEEK(7680+X+DX+44*(Y+DY))
220 IFY>1ANDY<6ANDP=32THEN420
230 IFY>6ANDP<>32ANDP<>35THEN420
240 IFY+DY=0ANDP<>32THEN420
250 IFY+DY=0ANDP=32THEN520
260 IFDY=0ANDDX=0THEN290
270 POKE7680+X+44*Y,OP:POKE38400+X+44*
Y,OC
280 OP=PEEK(7680+X+DX+44*(Y+DY)):OC=PE
EK(38400+X+DX+44*(Y+DY)):SC=SC+10
290 X=X+DX:Y=Y+DY:POKE7680+X+44*Y,35:P
OKE38400+X+44*Y,5
300 IFY=10RY=3THENX=X-1

```

```

310 IFY=20RY=4THENX=X+1
320 IFY>5ANDY<10THENPOKE7680+X+44*Y,32
330 POKE2,30:POKE1,44:SYS820:POKE1,88:
SYS866:POKE1,132:SYS820:POKE1,176:SYS8
66
340 POKE2,31:POKE1,8:SYS820:POKE1,52:S
YS866:POKE1,96:SYS820:POKE1,140:SYS866
350 P=PEEK(7680+X+44*Y):IFP<>32ANDP<>3
5THEN420
360 IFY>5ANDY<10THENPOKE7680+X+44*Y,35
:POKE38400+X+44*Y,5
370 IFSC>HITHENHI=SC
380 PRINT"T:TI$ "V"L
390 PRINT"[GIA]P:"SC"[SU]":PRINT"[10 D
ESTR]PM:"HI"[2 SU]"
400 IFTI>TXTHEN420
410 GOTO140
420 POKE7680+X+44*Y,OP:POKE38400+X+44*
Y,OC
430 POKE7680+DX+X+44*(Y+DY),35
440 PRINT"[HOME]TAB(227)"SPLAT!!"
441 FORT=254T0175STEP-1:POKE$1+2,T:POK
E38400+X+DX+44*(Y+DY),1:NEXT:POKE$1+2,
0
450 L=L-1:IFL=0THEN800
460 FORT=1T01000:NEXT:GOTO70
470 PRINT"[BNC][3 GIU'] ALTRO GIOCO[2
SPZ][S/N]?"POKE36879,14
480 POKE198,0
490 GETA$:IFA$="S"THEN60
500 IFA$<>"N"THEN490
510 POKE2,0:SYS2
520 POKE7680+X+44*Y,OP:POKE38400+X+44*
Y,OC
530 POKE7680+X+DX+44*(Y+DY),35:POKE384
00+X+DX+44*(Y+DY),5
540 FORT=1T030:FORG=200T0210:POKE$1,6+
T:NEXT:NEXT:POKE$1,0
550 FORT=1T01000:NEXT:SC=SC+1000
560 IFX=3THENF1=35
570 IFX=8THENF2=35
580 IFX=13THENF3=35
585 IFX=18THENF4=35
590 FT=FT+1:IFFT=4THENFT=0:F1=32:F2=32
:F3=32:F4=32:TX=TX-120:SC=SC+4000
595 IFTX<250THENTX=240
600 GOTO70
610 DATA165,1,133,251,165,2,24,105,120
,133,252,162,20,160,0,177
615 DATA1,72,200,177,1,136,145
620 DATA1,200,104,145,1,136,177,251,72
,200,177
625 DATA251,136,145,251,104,200,145,25
1,202,16
630 DATA226,96,165,1,133,251,165,2,24,
105,120,133
635 DATA252,162,20,160,20,177,1,72,200

```

```

,177,1
640 DATA136,145,1,200,104,145,1,136,17
7,251,72,200
645 DATA177,251,136,145,251,104,200,14
5,251
650 DATA136,136,202,16,224,96
655 FORT=820T0913:READK:POKET,K:X=X+K:
NEXT:IFX<>12468THENSTOP
660 DATA153,90,60,60,90,129,66,36,31,5
6,124,255
665 DATA127,231,36,24,0,128,64,252,254
,231,36,24
670 DATA126,70,71,255,255,231,36,24,25
5,,248,7,24,231,0,255

```

```

680 DATA255,255,255,255,255,255,1,0,25
5,255,255,255,255,51,50,204
690 POKE56,28:POKE52,28:POKE51,0:POKE5
5,0
695 FORT=0T0511:POKE7168+T,PEEK(32768+
T):NEXT
700 FORG=7448T07503:READK:POKEG,K:NEXT
:POKE36869,255:RETURN
800 POKE36865,160:POKE36879,8:PRINT"[C
LR]"TAB(205)"[CYN]FINE GIOCO"
810 FORX=160T038STEP-.2:POKE36865,X:NE
XT
820 GOTO470

```

**VC
20**

ROBOT

**GG
GG**

Dovete cercare di fermare i robot improvvisamente impazziti che Vi stanno distruggendo il muro, prima che riescano a raggiungere le fondamenta perche' in tal caso potranno distruggere anche la Vostra astronave. Al lancio del programma Vi verra' richiesto se intendete utilizzare il joystick oppure la tastiera e di seguito il livello al quale desiderate gareggiare. Una piccola strategia di gioco: quando cercate di colpire i robot con le Vostre bombe, mirate un po' piu' avanti di dove si trova in quel momento il robot, questo per compensare il tempo impiegato dall'ordigno per raggiungere il suolo ed il bersaglio.

DESCRIZIONE PROGRAMMA

ISTRUZIONI 2-4: inizializzazione variabili

ISTRUZIONI 5-10: gestione modifica set dei caratteri standard, mediante i valori letti dopo le parole chiave DATA dalle istruzioni 500 a 520.

ISTRUZIONI 12-70: ciclo principale del programma che si occupa di gestire i movimenti dei vari personaggi.

ISTRUZIONI 100-170: routine richiamata dal ciclo principale, adibita alla gestione dei movimenti.

ISTRUZIONI 200-210: routine di gestione del tipo di bersaglio colpito.

ISTRUZIONI 215-230: routine di gestione punteggi a fronte del tipo di bersaglio colpito.

ISTRUZIONI 300-350: gestione massimo punteggio ottenuto e richiesta nome del giocatore che lo ha conseguito.

ISTRUZIONI 355-390: gestione fine gioco e relativi messaggi di coniato.

ISTRUZIONI 600-627: routine gestione messaggio iniziale del gioco.

ISTRUZIONI 630-730: routine gestione secondo messaggio iniziale del gioco.

LISTA VARIABILI DEL PROGRAMMA

PROGRAMMA : ROBOT

VAR. : NUMERO ISTRUZIONE

A	4	10				
A\$(	3	4	301	302	320	325
	330	360				
B	4					
B\$(	320	330	610	620	625	
B(	3	13	150			
BB	55	700	710	720		
C	100	110	120			
CH	2	14	300	380		
CO	2	3	380	550	560	
D(	3	12	110			
E	140	150	160	305	335	345
FF	710	720				
G	30	32	35	65	300	301
	302	315	320	325	330	356
	360	640	650			
G\$(	375	380	385			
HZ(	3	4	30	300	301	315
	320	325	330	360		
I	5	10	22	25	356	360
J	310	315	320	325	330	340
	710					
K	303	350				
L	550	627				
LI	10	15	32	65	380	
M	320	330				
P	2	45	50	200		

```

PK : 40 60 110
PL : 40 60 150
PM : 2 45 50
PT : 55 205 215 220 225
Q : 2 22 25 50 130 170
: 200
R : 15 110 130
RL : 150 170
RM : 110 130
S : 15 150 170
S1 : 2 55 200 230 550 560
SC : 30 65 215 220 300 301
: 380
T : 200 205 210 225 230
TI : 35
TI$ : 15
V : 15 35
VO : 2 3 300 380 550 560
W : 15 25 35
X : 4 60
Y : 14 65
Z : 14 33 65

```

```

2 POKE52,28:POKE56,28:CLR:S1=36876:CH=
S1-7:CO=S1+3:VO=S1+2:PM=1:Q=30720:P=77
46
3 DIMA$(4,3),H$(4,3),D(4),B(4):POKECD,
8:POKEVD,15:PRINT"[CLR][BNC]ATTENDERE
PREGO."
4 FORA=0T04:X=X+800:FORB=0T03:A$(A,B)=
"VIC-20":H$(A,B)=X:X=X-50:NEXTB,A
5 FORI=7168T07679:POKEI,PEEK(I+25600):
NEXT
10 FORI=0T039:READA:POKE7448+I,A:NEXT:
LI=4
12 D(0)=-1:D(1)=22:D(2)=22:D(3)=1:D(4)
=-22
13 B(0)=-1:B(1)=-22:B(2)=-22:B(3)=1:B(
4)=22:GOSUB600
14 GOSUB630:Y=800:Z=Y:POKECH,255
15 PRINT"[CLR]":R=8032:S=R+21:W=8054:V
=10:TI$="000000":LI=LI-1:IFLI=0THEN300
22 FORI=0T021:POKE8164+I,39:POKE8164+I
+Q,5:NEXT
25 FORI=8163TOWSTEP-1:POKEI,35:POKEI+Q
,2:NEXT
30 PRINT"[HOME][BNC]PUNTI:[SIN.]SCTAB
(12)"HI:[SIN.]H$(G,0)
32 PRINT"VITE : "LITAB(12)"LIVEL: "G+1
33 PRINT" EXTRA VITA A "Z
35 IFINT(TI/60)>VTHENPOKEW,32:W=W+1:V=
V+6
40 GOSUB100:IFPK=39ORPL=39THENGOSUB550
:GOTO15

```

```

45 PM=PM+(2*PM*(P+PM<7746ORP+PM>7767))
50 POKEP,32:P=P+PM:POKEP,36:POKEP+Q,1
55 GOSUB700:IFBB=32THENGOSUB200:POKES1
,0:IFPT=39THEN15
60 X=RND(1):IFX>.5THENGOSUB100:IFPK=39
ORPL=39THENGOSUB550:GOTO15
65 IFSC=>ZTHENLI=LI+1:Y=Y+100*(G+1):Z=
INT(Z+(Y/3))
70 GOTO30
100 C=C-1:IFC=-1THENC=4
110 RM=D(C):PK=PEEK(R+RM):IFNOT(PK=35)
THEN130
120 C=C+1:C=-C*(C<5):GOTO110
130 POKER,32:R=R+RM:POKER,38:POKER+Q,1
140 E=E-1:IFE=-1THENE=4
150 RL=B(E):PL=PEEK(S-RL):IFNOT(PL=35)
THEN170
160 E=E+1:E=-E*(E<5):GOTO150
170 POKES,32:S=S-RL:POKES,38:POKES+Q,1
:RETURN
200 FORT=P+22T08163STEP22:POKET,37:POK
ET+Q,1:POKES1,230-(INT(T-7724)/22)
205 PT=PEEK(T+22):IFPT<>32THEN215
210 POKET,32:NEXT:RETURN
215 IFPT=38THENSC=SC+50:GOSUB550:GOTO2
30
220 IFPT=35THENSC=SC+1:GOTO230
225 IFPT=39THENPOKET,32:GOSUB550:RETUR
N
230 POKET,32:POKET+22,32:POKES1,0:RETU
RN
300 PRINT"[CLR]":POKEVD,0:POKECH,240:I
FSC<H$(G,3)THEN355
301 PRINT"[CLR][2 GIU']NOME PREGO (8 L
ETTERE)":INPUTA$(G,3):H$(G,3)=SC
302 IFLEN(A$(G,3))>8THEN301
303 PRINT"[CLR]":FORK=0T03
305 E=0
310 FORJ=0T02
315 IFH$(G,J)>H$(G,J+1)THEN340
320 M=H$(G,J):B$=A$(G,J)
325 H$(G,J)=H$(G,J+1):A$(G,J)=A$(G,J+
1)
330 H$(G,J+1)=M:A$(G,J+1)=B$
335 E=1
340 NEXTJ
345 IFE=0THEN355
350 NEXTK
355 PRINT"[2 GIU'] [5 DESTR]IRVS ONJ[CY
N]*****[GIU']"
356 PRINT"[6 DESTR]IRVS ONJ[ROSSJAL LI
VELLO[BNC]G+1[2 GIU']":FORI=0T03
360 PRINT"[2 DESTR]A$(G,I)TAB(13)H$(G
,I):NEXT
370 PRINT"[3 GIU'] [DESTR]ALTRO GIOCO (
S/N)?"
375 GETG$:IFG$=""THEN375

```

```

380 IFB$="S"THENSC=0:LI=4:POKEV0,15:PO
KECH,255:POKECO,8:GOTO14
385 IFB$="N"THENEND
390 GOTO375
500 DATA247,247,247,0,127,127,127,0,60
,126,255,85,255,126,60,0,
510 DATA0,0,20,8,28,28,28,8,60,213,189
,189,189,36,36,102
520 DATA255,255,195,195,195,195,255,25
5
550 POKES1,0:POKECO,42:FORL=15TO0STEP-
.15:POKEVO,L:POKES1+1,140+L*4
560 NEXT:POKECO,8:POKES1+1,0:POKEVO,15
:RETURN
600 PRINT"[CLR][2 GIU'] [ROSS][RVS ON]J
[RVS OFF][BNC]OYSTICK O[2 SPZ][ROSS][R
VS ON]T[RVS OFF][BNC]ASTIERA?"
610 GETB$:IFB$="J"THENPOKE0,0:PRINT"[5

```

```

GIU'] [3 DESTRI[RVS ON][GIA]TASTO FUOC
O SOLO"
620 IFB$="T"THENPOKE0,1:PRINT"[5 GIU']
[3 DESTRI[RVS ON][GIA]USA BARRA SPAZIO
"
625 IFB$<>"J"ANDB$<>"T"THEN610
627 FORL=1TO2000:NEXT:RETURN
630 PRINT"[CLR][2 GIU'] [CYN]IMMETTERE
LIVELLO(1-5)"
640 INPUT"[2 GIU'] (1 E' DIFFICILE[2 SP
Z])";G:G=G-1
650 IFG<0ORG>4ORG<>INT(G)THEN630
660 RETURN
700 IFPEEK(0)THENBB=PEEK(197):RETURN
710 J=PEEK(37137):FF=((JAND32)=0):BB=6
4
720 IFFFTHENBB=32
730 RETURN

```

**CM
64**

SUONO

**PP
PP**

Un simpatico piccolo programma da utilizzare per generare un effetto sonoro. E' da utilizzare al termine di una parte del programma che ha fatto rimanere in attesa chi lo sta utilizzando. In esso e' anche compresa la gestione della pressione di un tasto per continuare nell'utilizzo del programma.

DESCRIZIONE PROGRAMMA

ISTRUZIONE 10: viene caricato nella variabile S l'indirizzo iniziale del SINTETIZZATORE MUSICALE del COMMODORE 64 e vengono azzerati tutti i registri ad esso inerenti.

ISTRUZIONE 20: vengono caricati i valori delle note, viene modificato il registro inerente all'ATTACK/DECAY della prima voce.

ISTRUZIONE 30: loop che si occupa di generare l'effetto.

ISTRUZIONE 40: il suono viene ripetuto sino a quando non viene premuto un tasto.

ISTRUZIONE 50: attenuazione totale del volume del SINTETIZZATORE.

LISTA VARIABILI DEL PROGRAMMA

PROGRAMMA : SUONO

VAR. : NUMERO ISTRUZIONE

L	:	10				
Q\$	:	40				
S	:	10	20	30	40	50
X	:	30				
Y	:	30				

```

10 S=54272:FOR L=0 TO 24:POKE S+L,0:NE
XT

```

```

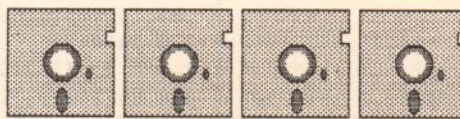
20 POKE S+1,68:POKE S+5,9:POKE S+24,15
30 FOR X=1 TO 10:POKE S+4,17:FOR Y=1 T
O 500:NEXT:POKE S+4,20:NEXT:PRINT"FINE
"

```

```

40 POKE S+4,21:GET Q$:POKE S+4,20:IF Q
$=""THEN 40
50 POKE S+24,0

```



VIC 20 DUMP GRAFICO VIC UU

Un interessante ausilio per trasportare mediante stampante, su carta quanto in quel momento e' presente sul video del VIC 20. Questo si puo' ottenere semplicemente digitando il programma sotto riportato, lanciandolo e premendo il tasto di funzione F1. Al lancio del programma. Vi verranno presentate due possibili scelte: caratteri minuscoli oppure caratteri maiuscoli. Premendo il tasto contrassegnato dalla lettera L, otterrete le minuscole, mentre premendo il tasto contrassegnato dalla lettera U, otterrete le maiuscole.

La solita raccomandazione: salvate sempre il programma, prima di effettuarne qualsiasi lancio!... In caso contrario correreste il rischio nell'eventualita' di aver commesso qualche errore di digitazione, di perderlo.

DESCRIZIONE PROGRAMMA

ISTRUZIONI 10-240: sono qui' contenuti i valori numerici da caricare in memoria e corrispondenti al programma in linguaggio, incaricato di svolgere la funzione di DUMP od HARD-COPY del video sulla stampante.

ISTRUZIONI 1000-1070: lettura dei valori contenuti nelle istruzioni prima viste e loro caricamento in memoria.

ISTRUZIONI 1080-1110: gestione richiesta di stampa a caratteri minuscoli o maiuscoli.

ISTRUZIONE 1120: istruzione alla quale si perviene mediante l'istruzione 310, nel caso di valori errati presenti dopo la parola chiave DATA, nelle istruzioni che vanno dalla 10 alla 240.

LISTA VARIABILI DEL PROGRAMMA

PROGRAMMA : DUMP GRAFICO VIC

VAR. : NUMERO ISTRUZIONE

A\$	:	1080
CS	:	1020 1030 1060
CS(	:	1000 1010 1060
EA	:	1010 1020 1040
F\$	:	1
I	:	1010 1020 1030 1040 1070
J	:	1020 1030 1040 1050
K	:	1030

L	:	1010 1060 1070 1120
SA	:	1010 1020 1080 1090 1100 1110
VEC	:	1100
X	:	1000 1010 1100 1110
Y	:	1000 1010 1100 1110

1 F\$="VICSCRN"

10 DATA 907,869,1199,1261,1403,1782,1699,1949

20 DATA 1343,1366,1652,1948,1757,1715,1366,1219

30 DATA 1593,1531,1895,1902,704

40 DATA 120,173,20,3,141,54,3,173,21,3,141,55

50 DATA 3,169,30,141,20,3,169,0,141,21,3,169

60 DATA 0,141,56,3,88,96,165,197,201,39,208,5

70 DATA 173,56,3,240,3,108,54,3,169,255,141,56

80 DATA 3,169,8,162,4,160,255,32,186,255,169,0

90 DATA 32,189,255,32,192,255,162,8,32,201,255,169

100 DATA 0,133,251,133,253,169,30,133,252,162,23,160

110 DATA 21,177,251,136,240,4,201,32,240,247,200,200

120 DATA 140,53,3,160,0,177,251,142,52,3,170,192

130 DATA 0,208,15,169,16,32,210,255,169,50,32,210

140 DATA 255,169,57,32,210,255,138,201,128,144,22,41

150 DATA 127,170,165,253,208,28,169,255,133,253,169,18

160 DATA 32,210,255,56,176,16,24,144,186,165,253,240

170 DATA 9,169,0,133,253,169,146,32,210,255,138,201

180 DATA 32,144,13,201,64,176,13,32,210,255,174,52

190 DATA 3,24,144,15,105,64,144,243,201,96,176,4

200 DATA 105,32,144,235,24,144,241,200,204,53,3,208

210 DATA 152,169,13,32,210,255,169,22,24,101,251,133

220 DATA 251,144,2,230,252,202,208,182,169,13,32,210

230 DATA 255,32,210,255,32,204,255,169,8,32,195,255

240 DATA 169,0,141,56,3,76,24,235

```

1000 DIMCS(21):POKE56,PEEK(56)-1:X=PE
EK(55):Y=PEEK(56)
1010 FORI=1TO21:READCS(I):NEXT:SA=X+2
56*Y:EA=SA+247:L=1
1020 FORI=SATOEASTEP12:CS=0:FORJ=0TO1
1
1030 READK:POKEI+J,K:CS=CS+K
1040 IFI+J=EATHENJ=11:I=EA
1050 NEXTJ
1060 IFCS<>CS(L)THEN1120
1070 L=L+1:NEXTI

```

```

1080 PRINT"[6IU']MAIUSCOLE O MINUS.";
:INPUT"[3 SP2IU[3 SIN.]";A$:IFA$="L"
HENPOKESA+54,7
1090 PRINT"INDIRIZZO INZ."="SA
1100 POKESA+78,PEEK(648):VEC=SA+30:Y=
INT(VEC/256):X=VEC-Y*256
1110 POKESA+14,X:POKESA+19,Y:SYSSA:NE
W
1120 PRINT"ERRORE IN DATA,ISTRUZIONE#
";40+(L-1)*10

```

CM INTERRUPT UU 64 UU

La possibilità di far muovere in maniera indipendente delle SPRITE sul video e' richiesta nella realizzazione di giochi ad un certo livello.

Questo piccolo programma si occupa appunto di gestire tale possibilità, facendo muovere in maniera del tutto indipendente una SPRITE sul video, in senso orizzontale. Una volta attivata, la SPRITE continuerà a muoversi ed in tal modo verrà, tolto al programma BASIC che si occupa di gestire il gioco, l'onere della sua gestione. Il programma e' facilmente inseribile ove si presenti questa necessità.

DESCRIZIONE PROGRAMMA

Istruzioni 10-40: gestiscono la creazione della SPRITE utilizzata dalla routine in linguaggio macchina, adibita alla gestione della stessa in maniera indipendente.

Istruzioni 100-120: i valori in esse riportati servono alla costruzione in memoria della SPRITE.

Istruzione 510: carica in memoria la parte in linguaggio macchina situata dall'istruzione 520 all'istruzione 550 ed adibita alla gestione indipendente della SPRITE, creata nelle precedenti istruzioni. In essa la medesima routine viene anche attivata mediante: SYS 49152.

Istruzioni 520-550: vi si trovano i valori della parte in linguaggio macchina adibita alla gestione della SPRITE.

LISTA VARIABILI DEL PROGRAMMA

PROGRAMMA : INTERRUPT

VAR.	NUMERO	ISTRUZIONE
A	10	510
T	10	510
V	20	30

INTERRUPT

```

5 REM
10 FOR T=12736 TO 12798:READ A:POKE T,
A:NEXT
20 V=53248:POKE 2047,199:POKE V+21,128
:POKE V+46,12:POKE V+14,0:POKE V+15,55
30 POKE V+29,128
40 POKE 251,5:POKE 252,3
100 DATA 0,0,0,0,48,0,1,248,0,3,248,0,
7,254,0,31,255,0,63,254,0,63,255,0
110 DATA 31,255,192,127,255,192,255,25
5,192,255,255,128,127,255,128,31,254,0
120 DATA 15,254,0,15,252,0,1,240,0,0,2
40,0,0,96,0,0,0,0,0,0,0
500 REM
510 FOR T=49152 TO 49229:READ A:POKE T
,A:NEXT:SYS 49152
520 DATA 120,169,13,141,20,3,169,192,1
41,21,3,88,96,198,251,165,251,240,3,76
530 DATA 74,192,173,16,208,41,128,240,
23,173,14,208,201,101,208,31,173,16,20
8
540 DATA 37,127,141,16,208,169,0,141,1
4,208,76,67,192,173,14,208,201,255,208
550 DATA 8,173,16,208,5,128,141,16,208
,238,14,208,165,252,133,251,76,49,234,
0

```



Tutti i lettori che desiderano comunicare con il mercato, possono approfittare della rubrica di piccoli annunci gratuiti, compilando e spedendo a: CO.STA EDIZIONI - VIA MAZZINI 128 - 50100 PISA il sottostante tagliando.

I _ I VENDO				I _ I COMPRO				I _ I SCAMBIO				I _ I INFORMO			
I _ I HARDWARE				I _ I SOFTWARE				I _ I ALTRE							

NOME _____															
VIA _____ CITTA' _____															
C.A.P. _____ TELEFONO _____ ORARIO _____															

VENDO COMMODORE 64 piu' stampante MPS801, piu' registratore C2N, joystick e oltre 30 giochi a lire 1.000.000. Inoltre regalo un pacco da 2.000 fogli per stampante, alcuni volumi ed eventualmente anche televisore B/N. Scrivere o telefonare a:
TANFANI UGO - Via Monti, 1 - 20090 TREZZANO s/N - (MI)
Tel. 02/4455748.

SCAMBIO - VENDO programmi per COMMODORE 64, preferibilmente su disco - ANNUNCIO SEMPRE VALIDO - Inviatemi le vostre liste.
SICCARI ROBERTO - Via Matteotti, 178 - 18038 SANREMO
Tel. 0184/ 75687 - ORARIO: 15 -19

VENDO - VIC 20 + registratore + 7 libri d'informatica + 5 cassette di giochi di cui 2 cassette d'istruzioni. PREZZO L.200.000.
DE STEFANO CESARE - Via Napoli, 81 - 80053 - C/MARE DI STABIA - Tel. 081/ 8722007 - ORARIO: 20.00 - 22.00

VENDO - SCAMBIO - PROGRAMMI per COMMODORE 64 su dischetti e cassette.
Scrivere a:
TAGLIACCOZZO ARIEL - Via Sebino, 16 - 00199 - ROMA

VENDO - SCAMBIO - PROGRAMMI per il COMMODORE 64 (oltre 470) sia su nastro sia su disco. Ultime novita' per cassetta.

Scrivere o telefonare a:
MARANGONI ANDREA - Via Pausula, 72 - 62014 - CORRIDONIA
Tel. 0733/ 292416 - ORARIO: pasti.

VENDO - COMPRO - SCAMBIO - PROGRAMMI per il COMMODORE 64. Tra gli altri, un eccellente TOTOCALCIO in linguaggio macchina. Accetta qualsiasi tipo di condizionamenti. Scrivere o telefonare a:
STOLZI LIBERO - Via S.Maria, 1 - 53021 - ABBADIA S.S. (SIENA) - Tel. 0577/ 848117 - ORARIO: 12.00 - 13.00.

CO.STA MARKETING

A tutti gli amici lettori di SOFTWARE DOSSIER un'offerta a prezzi molto vantaggiosi di prodotti di consumo per personal e home computer. Per effettuare l'ordinazione è sufficiente compilare e spedire in busta chiusa a:

CO.STA SRL - VIA MAZZINI 128 - 56100 PISA - SERVIZIO MARKETING

Il sottostante tagliando, allegando il pagamento dell'importo complessivo in contanti o tramite assegno bancario o vaglia postale. I prezzi sono comprensivi di IVA e oneri di spedizione. Per spedizioni in contrassegno supplemento fisso di lire 3.000.

DISCHETTI

☐ **Cod. V001**

Minifloppy 5" 1/4 semplice faccia, semplice densità.
Scatola da 10 dischetti **Lire 45.000**
Ogni scatola oltre la 1ª sconto 5%.

☐ **Cod. V002**

Minifloppy 5" 1/4 doppia faccia, doppia densità.
Scatola da 10 dischetti **Lire 55.000**
Ogni scatola oltre la 1ª sconto 5%.

☐ **Cod. 8002**

Floppy 8" doppia faccia, doppia densità. Scatola da 10 dischetti **Lire 65.000**
Ogni scatola oltre la 1ª sconto 5%.

CONTENITORI

☐ **Cod. BF01**

Contentore a scatola FLEX BOX per 10 minifloppy con coperchio Rosso, Giallo o Nero **Lire 5.000**



Cod. C040

Contentore in plastica da 40 minifloppy, con serratura **Lire 28.000**



Cod. C090

Contentore in plastica da 90 minifloppy, con serratura **Lire 35.000**

☐ **Cod. TD01**

Supporto per stampante e contenitore di moduli continui TOP DESK in plastica trasparente **Lire 90.000**

FLOPPY DISKS

☐ **Cod. C001**

Floppy Disks Album **Lire 8.500**

☐ **Cod. E001**

Buste per 2 Floppy Disks 5" 1/4 con etichette in clear antimagnetico **Lire 800**

☐ **Cod. E002**

Buste per Floppy Disks 8" con etichetta in clear antimagnetico **Lire 800**

Ordine minimo lire 50.000

CODICE	QUANTITÀ	PREZZO TOTALE	
			Nome _____
			Cognome _____
			Via _____ N° _____
			Città _____
			_____ CAP _____
			Firma _____
TOTALE L.			



PERSONAL COMPUTER

CLUB

GIOCHI-CONCORSO

PROGRAMMI E TRUCCHI

PROVE HARDWARE E SOFTWARE

PROGETTI SOFTWARE

BIBLIOTECA PERSONALE

e le rubriche

TUTTOCOMMODORE

TUTTOSINCLAIR

TUTTOMSX

**OGNI MESE
IN EDICOLA
LIRE 3.000**